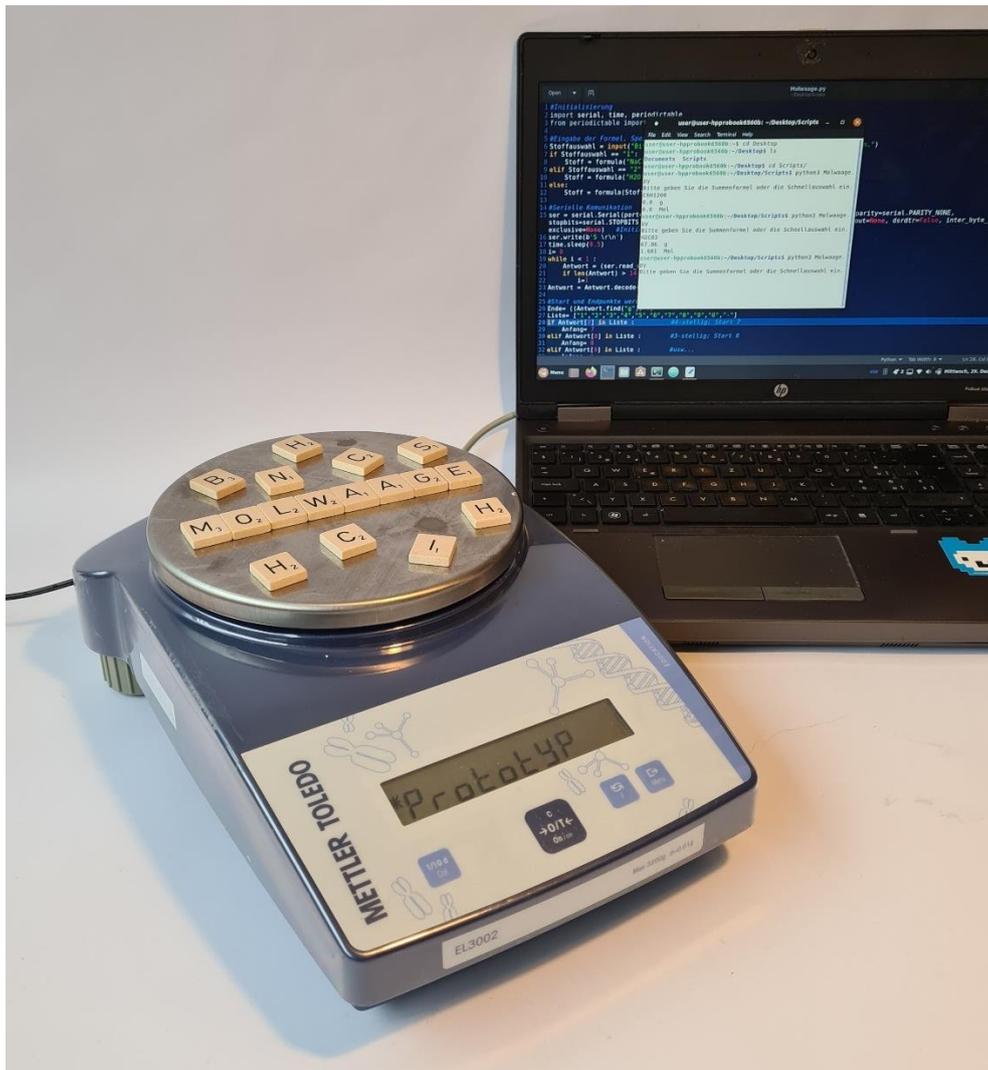


Molwaage

Entwicklung eines Prototyps



Maturaarbeit von Felix Brugnoni
Betreut durch Dr. Rainer Steiger
Abgegeben im Dezember 2021
Kantonsschule Schaffhausen

INHALTSVERZEICHNIS

1	Einleitung.....	3
1.1	Einführung.....	3
1.2	Zielsetzung.....	3
1.3	Anforderungen und Einsatz	3
1.3.1	Einsatzszenario	3
1.3.2	Muss-Anforderungen	4
1.3.3	Kann-Anforderungen	4
2	Lösungsvarianten	6
2.1	Einführung.....	6
2.2	Grundlegendes	6
2.3	Variantenübersicht	6
2.4	Beurteilung	6
2.4.1	Machbarkeit.....	6
2.4.2	Weiterentwicklung	7
2.5	Variantenentscheid	8
3	Vorgehen und Entstehung.....	9
3.1	Einführung.....	9
3.2	Verwendete Werkzeuge.....	9
3.2.1	Programmiersprache	9
3.2.2	Hardware.....	10
3.2.3	Schnittstelle	10
3.2.4	Betriebssystem	10
3.2.5	Entwicklungsumgebung.....	10
3.2.6	Serieller Monitor	11
3.2.7	Bibliotheken.....	11
3.3	Entwicklungsvorgang	12
3.3.1	Allgemein.....	12
3.3.2	Hilfen bei der Entwicklung	12
3.4	Leistungen und Resultate	13
4	Lösungsbeschreibung.....	15
4.1	Einführung.....	15
4.2	Kommunikation Waage – PC	15
4.2.1	Einstellungen.....	15

4.2.2	Nachricht auf Waage anzeigen	16
4.2.3	Gewichtsabfrage.....	17
4.3	Mol-Rechnung	18
4.3.1	Aufruf einer molaren Masse.....	19
4.3.2	Kurzbefehle	19
4.3.3	Formeleingabe	20
4.3.4	Stoffmengenberechnung	20
4.4	Bearbeitung des Rückgabewertes / String-Manipulation	20
4.4.1	Decodierung.....	20
4.4.2	String-Manipulation 1: Endpunkt.....	21
4.4.3	String-Manipulation 2: Startpunkt.....	21
4.4.4	String-Manipulation 3: Aufruf eines Abschnitts.....	22
4.5	Übersicht.....	23
4.5.1	Initialisierung: Aufruf der Bibliotheken.....	23
4.5.2	Anzeigen der Masse.....	23
4.5.3	Anzeigen der Stoffmenge	23
4.5.4	Vorkehrungen.....	23
4.5.5	Anzeige	24
5	Nachüberlegungen	25
5.1	Endprodukt	25
5.2	Weiterentwicklung.....	25
5.3	Entstehung	26
5.4	Weitere Gedanken	26
5.5	Schlusswort	27
6	Anhang.....	28
6.1	Abkürzungsverzeichnis.....	28
6.2	Bibliografie	28
6.3	Abbildungsverzeichnis.....	29
6.4	Ressourcen bei der Entwicklung	29
6.5	Programmcode	29
6.6	Dokumentinfo	31

1 Einleitung

1.1 Einführung

Die erste Idee für diese Arbeit gelangte im Jahr 2019 zu mir. Thomas Stamm, dessen Unterricht ich im Chemie-Schwerpunktfach geniessen darf, äusserte einen Vorschlag für eine Maturaarbeit. Dabei handelte es sich um eine Molwaage, ein Objekt, welches im Unterricht zum Thema Mol, Avogadro und Stoffmengen eingesetzt würde. Dieses sollte, wie der Name schon andeutet, eine Waage sein, welche zu Anschauungszwecken die Stoffmenge einer Probe in Mol anzeigt.

Aus eigener Erfahrung kannte ich damals bereits die Schwierigkeit beim genannten Thema. Stoffmengen, molare Massen und Avogadro schienen mir sehr abstrakt und nur dank Gewöhnung, viel Zeit und genügender Thematisierung konnte ich mir irgendwann etwas Konkretes darunter vorstellen oder chemisch damit arbeiten. Somit hielt ich Herrn Stamms Idee für sinnvoll und nahm sie in mein Repertoire an möglichen Maturarbeitsthemen auf. Nicht nur des technischen Interesses, sondern auch der Faszination für Wissensvermittlung wegen, blieb die Molwaage ein Favorit.

Anfangs dieses Jahres, als die Zeit reif für den Entscheid zum MA-Thema war, erinnerte ich mich an die Molwaage. Immer nach Überschneidungen von Technik oder Informatik mit Naturwissenschaften wie Biologie oder Chemie suchend, schien dies eine interessante Herausforderung zu sein.

1.2 Zielsetzung

Ziel dieser Maturaarbeit ist das Herangehen an die von Herrn Stamm vorgestellte Idee, das Finden von technischen Möglichkeiten und Konzepten und nach der Evaluierung ebendieser die Umsetzung in einen Prototyp.

Nebenbei erhoffe ich mir sowohl meine technischen Kenntnisse und Programmier-Fähigkeiten zu erweitern wie auch Erfahrungen in der Projekt-Handhabung und dem zielorientierten Vorgehen zu sammeln.

Das vorliegende Dokument beschreibt die Hintergründe und die Vorgehensweise der Entstehung sowie die Details und Besonderheiten des entstandenen Prototyps. Das Resultat wird am Ende präsentiert, bewertet und mögliche zukünftige Weiterentwicklungen werden besprochen.

1.3 Anforderungen und Einsatz

Die Molwaage war in der ursprünglichen Idee nur im Ansatz beschrieben. Die Angaben von Herrn Stamm waren sinngemäss: Eine Waage oder ein beliebiges Gerät, das eine Waage begleitet, welches die Stoffmenge einer Probe in Mol anzeigt.

Die Stoffmenge einer Probe hängt direkt von der Masse, oder dem Gewicht, dieser Probe ab. Weiter variiert die Umrechnung mit dem Stoff der Probe. Wie die Waage oder das Gerät (bezüglich Auswahl des Stoffs) bedient oder das Resultat angezeigt werden soll, hat Herr Stamm nicht geäussert. Da hier viel Spielraum bleibt, können selbst mögliche Lösungsansätze gesucht, verglichen und umgesetzt werden. Um die Fragen nach der Bedienung und den daraus folgenden nötigen technischen Einzelheiten zu beantworten, werden im Folgenden einige mögliche Szenarien umschrieben, unter denen die Anwendung einer Molwaage vorstellbar ist.

1.3.1 Einsatzszenario

Folgenden Rahmenbedingungen, bezüglich der Unterrichtssituation in der die Molwaage eingesetzt wird, werden angenommen:

- Die Molwaage wird unter anderem während der Thematisierung des Unterrichtsblockes, welches das Thema Mol, Avogadro usw. beinhaltet, eingesetzt.

- Die Waage dient als Begleitung zu der im oben genannten Unterrichtsteil vorkommenden Theorie wie auch zur Veranschaulichung oder Vertiefung.
- Ebenfalls kann zu anderen beliebigen Zeitpunkten auf die Molwaage zurückgegriffen werden, beim Wiederauftauchen der Themata oder zu beliebigen Versuchen, bei denen die Lehrperson einen Nutzen der Molwaage sieht.
- Entweder nutzt die Lehrperson die Molwaage im Frontalunterricht, bei einem Versuch (möglicherweise im Labor) oder zum selbstständigen Bedienen durch Schülerinnen und Schüler.

Weiter lassen sich einige Szenarien der schulischen Nutzung und die ihnen zugehörigen Arten der Präsentation unterscheiden:

- Im Frontalunterricht befindet sich die Waage am Pult der Lehrperson und die S.u.S. blicken von deren Plätzen aus entweder auf eine spezielle Anzeige oder die Resultate werden mit der vorhandenen Leinwand und dem Beamer des Theoriezimmers projiziert.
- Sind bei einem Versuch die S.u.S. nahe an der Molwaage selbst, so reicht entweder das integrierte oder ein (kleines) externes Display.
- In einer Laborsituation oder beim Umgang der S.u.S. mit der Molwaage reicht ebenfalls das integrierte oder ein (kleines) externes Display.

Technisch gesehen unterscheiden sich die obigen Szenarien nur unerheblich. Eine technische Lösung für ein Szenario kann mit Anpassungen leicht in einem anderen Szenario genutzt werden, da die Grundfunktionsweise unverändert bleibt.

Die Aufteilung in Muss- und Kann-Anforderungen ermöglicht einen Fokus auf die wichtigsten Bereiche bei der Entwicklung und stellt später die Grundbausteine für das Vorgehen. Muss- und Kann-Anforderungen sind gewissermassen selbsterklärend, aber einige spezielle Vorteile einer solchen Trennung gibt es bei diesem Projekt. Die Anforderungen werden im Folgenden aufgestellt und ihr Zweck beschrieben.

1.3.2 Muss-Anforderungen

Folgende Anforderungen muss der zu produzierende Prototyp der Molwaage erfüllen:

- M-1 Eine Auswahl von mindestens drei Stoffen kann mit der Molwaage gewogen und automatisch umgerechnet werden,
- M-2 Mindestens eine Art der visuellen Ausgabe des ermittelten Wertes ist vorhanden.

Diese Muss-Anforderungen werden die Grundlagen dieser Arbeit sein und stellen Grundbausteine für eine mögliche Weiterentwicklung dar. In anderen Worten handelt es sich um ein Proof of Concept: Sind die Aspekte erfüllt, dann sind die grössten Probleme gelöst und der Weg ist frei für die Weiterentwicklung.

1.3.3 Kann-Anforderungen

Nachdem wie oben beschrieben das Proof of Concept irgendwann im Entstehungsprozess erfüllt sein wird, sollen Weiterentwicklungen folgen. Sie sind technisch weniger wichtig, aber beispielsweise für die Nutzbarkeit im Unterricht nötig und sollen deshalb möglichst erfüllt werden. Ein Teil dieser Kann-Anforderungen wird womöglich fast von selbst erfüllt sein, andere können mit mittlerem Aufwand erfüllt werden und einzelne würden den Rahmen dieser Arbeit sprengen und deshalb ausserhalb davon nachgeliefert werden.

- K-1 Einfacher Betrieb der Molwaage,
- K-2 Eine größere Auswahl an zu testenden Stoffen,
- K-3 Das Eingeben beliebiger Stoffformeln,
- K-4 Anzeige des Resultates an einem Display in einfacher, ungeschöner Form,

- K-5 Als weitere Stufe von obigem die geschönte, gut leserliche Präsentation,
- K-6 Vorkehrungen zum Verhindern von Programmabstürzen,
- K-7 Ausführbarkeit der Anwendung auf jedem PC, auch ohne Installation von Hilfsprogrammen (z.B. Python) oder Reproduzierbarkeit bei proprietären Lösungen,
- K-8 Ausführbarkeit der Anwendung unabhängig vom installierten Betriebssystem,
- K-9 Der Einsatz von auf der Waage integrierten Tasten und/oder des Displays.

2 Lösungsvarianten

2.1 Einführung

Nachdem im vorangehenden Kapitel die nötigen Vorüberlegungen und Zielsetzungen gemacht wurden, wird im nächsten Schritt eine technische Diskussion im Sinne der Vorbereitung für die Umsetzung geführt. Ihr Ziel ist es eine erste Idee über den Umfang der zu erarbeitenden technischen Umsetzung zu erhalten und auch einige wegweisende Vergleiche und Entscheide zu fällen.

2.2 Grundlegendes

Die Wahl der Waage steht mit der Mettler-Toledo EL3002 fest. Die folgend diskutierten Varianten müssen daher alle eine serielle Kommunikation (RS232) mit der Waage ermöglichen. Dies resultiert in Hardware- wie auch Software-artigen Anforderungen an die weiteren beteiligten Geräte.

2.3 Variantenübersicht

Für die Variantenbildung sind folgende Punkte zu beachten:

- Als Eingabe steht, neben der Waage, eine integrierte oder externe Tastatur zur Auswahl.
- Prinzipiell stehen als Rechner ein Laptop / Desktop oder ein Microcomputer (Raspberry Pi oder Arduino) zur Auswahl (Rechner bedeutet das verarbeitende Gerät).
- Als Display stehen, je nach Rechner, ein integrierter Bildschirm, ein externer Bildschirm oder ein Beamer zu Verfügung.

Vorerst wird auf die möglichen Betriebssysteme nicht eingegangen. Diese werden im Kapitel 3 angesprochen.

Daraus ergeben sich folgende Varianten:

Rechner	x.1: Display	x.2: Beamer
1.x: Laptop / Desktop	1.1a: Laptop mit RS232-Schnittstelle	1.2a: Laptop oder Desktop mit RS232-Schnittstelle, an Beamer
	1.1b: Laptop plus RS232-Adapter	1.2b: Laptop oder Desktop plus RS232-Adapter, an Beamer
2.x: Microcomputer	2.1a: Raspberry Pi plus Display und Tasten	2.2a: Raspberry Pi plus Tasten, an Beamer
	2.1b: Arduino plus Display und Tasten	2.2b: Arduino plus Tasten, an Beamer

Table 1: Variantenübersicht

2.4 Beurteilung

Für die Beurteilung der Varianten werden einige Eigenschaften aufgelistet und die voraussichtliche Eignung zur Weiterentwicklung verglichen.

2.4.1 Machbarkeit

Was die Einfachheit der Entwicklung angeht, haben die verschiedenen Varianten Vor- und Nachteile, die folgend aufgelistet sind.

- Die Entwicklung auf Desktop oder Laptop ist vorteilhaft, weil sie die Programmierung und das Testen von Programmen auf das gleiche Gerät bringt.
- Die Mobilität einer Laptop-Lösung ist für die Entwicklung und später für den Betrieb vorteilhaft.
- Microcomputer sind für die Entwicklung von Prototypen ausgelegt, die zugehörige Hard- und Software ist weit verbreitet und gut zugänglich.
- Aus den genannten Rechnern hat Arduino die geringste Leistungsfähigkeit. Für komplexe, hochauflösende grafische Darstellung ist er höchstwahrscheinlich nicht geeignet.
- Alle Varianten mit Desktop oder Laptop lassen sich auch im Nachhinein ineinander überführen. Sie sind kompatibel. Ein Displaywechsel wie auch das Wechseln der Tastatur sind ohne Anpassung in der Software möglich.
- Die Auswahl des Betriebssystems kann die Funktionalität und die Kompatibilität der Gesamtlösung / des Prototyps beschränken.
- Microcomputer ermöglichen eine einfache Software-Installation, im Vergleich zu Laptop / PC, und die Materialkosten sind tiefer.

2.4.2 Weiterentwicklung

Die Bewertung berücksichtigt die Erreichbarkeit einzelner Kann-Anforderungen und deren Wichtigkeit. Die Grundlage für die Einschätzungen besteht aus Vorwissen und teilweise kleinen Recherchen zu den vorgestellten Varianten. Daraus wird abgeschätzt, wie schwer die aufgeführten Aspekte je Variante zu erreichen sind.

	Gewichtung	1.1a	1.1b	1.2a	1.2b	2.1a	2.1b	2.2a	2.2b
K-1 Einfacher Betrieb der Mollwaage	2	2	1	2	1	2	2	1	1
K-2 Eine grössere Auswahl an zu testenden Stoffen	2	2	2	2	2	1	1	0	0
K-3 Das Eingeben beliebiger Stoffformeln	2	2	2	2	2	1	1	0	0
K-4 Anzeige des Resultates an einem Display in einfacher, ungeschöner Form	2	2	2	2	2	2	2	1	1
K-5 Als weitere Stufe von obigem die geschönte, gut leserliche Präsentation	1	1	1	1	1	1	1	0	0
K-6 Vorkehrungen zum Verhindern von Programmabstürzen	2	2	2	2	2	1	1	1	1
K-7 Ausführbarkeit der Anwendung auf jedem PC auch ohne Installation von Hilfsprogrammen (z.B. Python) Beziehungsweise Reproduzierbarkeit	1	1	1	1	1	2	2	2	2

	Gewichtung	1.1a	1.1b	1.2a	1.2b	2.1a	2.1b	2.2a	2.2b
K-8 Ausführbarkeit der Anwendung unabhängig vom installierten Betriebssystem	1	0	0	0	0	n.a.	n.a.	n.a.	n.a.
K-9 Der Einsatz von auf der Waage integrierten Tasten und / oder des Displays.	1	2	2	2	2	2	2	2	2
Punktzahl	--	22	21	22	21	19	19	12	12

Tabelle 3: Auflistung der Anforderungen mit Zuordnung zu den Varianten [n.a.; not available; nicht angegeben]

Gewichtung:		Schwierigkeit:	
2	Sehr wichtig	2	Machbar
1	Wichtig	1	Eher schwierig
0	Unwichtig	0	Sehr schwierig

2.5 Variantenentscheid

Die Varianten 2.1b und 2.2b, basierend auf Arduino-Microcomputer, müssen ausgeschlossen werden. Das Arduino-System, wäre zwar für Projekte dieser Art geeignet, erzielt aber aus verschiedenen Gründen eine tiefere Punktzahl. Hauptgründe sind die eingeschränkte Leistung und vor allem die fehlenden Anschlüsse (Bildschirm-Ausgänge).

Zwischen Laptop und Desktop besteht, wie erläutert, kein erheblicher Unterschied. Es läuft also auf die Wahl zwischen Raspberry Pi und Laptop / Desktop hinaus. Beide haben grosse Vorteile. Der Raspberry Pi liegt in der obigen Bewertung um wenige Punkte zurück.

Weiter würde der Entscheid zum Raspberry Pi noch den Erwerb eines solchen sowie das Gelangen an die benötigten Schnittstellenadapter nach sich ziehen, während privat schon ein Laptop mit integrierter RS232-Schnittstelle vorhanden ist.

Damit steht der Entscheid fest: Die Entwicklung des Prototyps erfolgt mit der Variante 1.1a oder 1.2a auf einem Laptop mit integrierter RS232-Schnittstelle. Ob ein internes / externes Display oder ein Beamer verwendet wird ist technisch nicht so bedeutend.

Allerdings kann sich die Variante während der Entwicklung noch ändern, sollte sich die Umsetzung mit dem erwähnten Laptop als unmöglich herausstellen.

3 Vorgehen und Entstehung

3.1 Einführung

Dieses Kapitel klärt den Entstehungsprozess. Die Details zum Entwicklungsvorgehen folgen nach einer Erläuterung der verwendeten Werkzeuge. Schliesslich wird noch ein zeitlicher Überblick gegeben. Die inhaltliche und technische Entwicklung wird noch nicht angesprochen, dies geschieht in Kapitel 4.

3.2 Verwendete Werkzeuge

Im Folgenden werden Soft- und Hardware-Werkzeuge und Hilfsmittel vorgestellt.

3.2.1 Programmiersprache

Mehrere Programmiersprachen kommen für ein Projekt wie das vorliegende in Frage. Da das Erreichen der Muss-Anforderungen oberste Priorität hat, muss eine zugängliche Programmiersprache gewählt werden, mit dem Ziel, dass diese nicht noch in ihrer Komplexität eine zusätzliche und unnötige hohe Hürde darstellt.

Trotzdem soll eine ausreichend komplexe Sprache gewählt werden. Die höhere Programmiersprache Python dürfte passen. Python's gut verständliche Struktur und Syntax erlauben im Vergleich zu anderen Programmiersprachen eine intuitivere Programmierung und gute Lesbarkeit. So müssen in Python für die gleiche Funktion weniger Zeilen geschrieben werden und die Zusammenhänge sind einfacher zu verstehen. Nichtsdestotrotz bleibt Python aber sehr leistungsstark und ist weit verbreitet und vielerorts im Einsatz. Python ist also viel mehr als nur eine reine Übungs- oder Einsteiger-Programmiersprache.

Übersetzt und angepasst, von der offiziellen Python-Website:¹

Was ist Python? Eine Zusammenfassung

Python ist eine interpretierte, objektorientierte höhere Programmiersprache mit dynamischer Semantik. Seine hochentwickelten, integrierten Datenstrukturen, kombiniert mit dynamischer Typisierung und dynamischer Bindung, machen es sehr attraktiv für die schnelle Anwendungsentwicklung sowie für die Verwendung als Skript- oder Klebesprache, um bestehende Komponenten miteinander zu verbinden. Der einfache, leicht zu erlernende Syntax von Python betont die Lesbarkeit und reduziert somit die Kosten für die Programmpflege. Python unterstützt Module und Pakete, was die Modularität von Programmen und die Wiederverwendung von Code fördert.

Wie in der obigen Beschreibung angedeutet ist ein Vorteil von Python das Vorhandensein von vielen Programm-Bibliotheken. Dank diesen kann beim Programmieren direkt auf die spezifischen Fragen und Fokuspunkte dieses Projekts eingegangen werden. Beispielsweise müssen die Funktionen zur seriellen Kommunikation nicht neu programmiert werden, sondern stehen zur Verfügung und können als Bausteine für ein zielgerichtetes Programm verwendet werden, bei dem die für die Molwaage speziellen Fragen im Zentrum stehen.

Es wird die Python-Version 3.7 verwendet. Es gibt bereits neuere Versionen, die jedoch experimentell, weniger stabil und weniger weit verbreitet sind.

¹ python.org (Python Software Foundation): <https://www.python.org/doc/essays/blurbl/>, Stand: 11.12.2021

3.2.2 Hardware

Mit der Mettler-Toledo EL3002 wurde eine schulische Waage mit RS232-Schnittstelle vorgegeben, welche weder topmodern noch technisch veraltet ist. Der Hersteller Mettler-Toledo scheint wie privat beobachtet eine hohe Bekanntheit und grosses Ansehen zu geniessen. Sehr erfreulich sind die verfügbaren Dokumentationen zum Umgang mit der Schnittstelle, in denen die Befehle und Antworten detailliert beschrieben sind.



Abbildung 1: Die Waage

Wie im Kapitel 2 erwähnt, ist privat ein Laptop mit integrierter RS232-Schnittstelle vorhanden. Dies schliesst bereits mögliche Probleme mit Adaptern aus und bietet eine portable Lösung. Es handelt sich bei dem Laptop um ein Modell der Marke Hewlett-Packard, der Modellname lautet ProBook 6560b. Für eine Anwendung im Unterricht würde sich zwar ein Laptop mit integrierter Schnittstelle besonders gut eignen, aber der erwähnte dient nur für den Prototypen und die Entwicklung. Auf dem Titelblatt ist der Laptop in einem beispielhaften Aufbau mit der Waage zu sehen.

3.2.3 Schnittstelle

RS232 ist ein Schnittstellenstandard für serielle Schnittstellen. Er legt verschiedene Einzelheiten fest, damit zwei Geräte problemlos kommunizieren können. In früheren Jahren konnte der Standard mit dem ihm zugehörigen VGA-ähnlichen Stecker in Druckern oder Modems angetroffen werden, heute wird er im Privatgebrauch aber kaum noch verwendet. In Labors wird er noch immer eingesetzt. (Diese Beschreibung wurde auf einen Artikel von rs-online gestützt.²)

3.2.4 Betriebssystem

Sowohl Windows als auch Linux sind dem Autor bekannt und für die Entwicklung geeignet. Linux wurde aber aufgrund einiger Vorteile ausgewählt. Linux ist gratis und einfach zu installieren, vielseitig, gut konfigurierbar und unterstützt viele Programmiersprachen.

Jedoch sollte nicht eine komplex zu bedienende Linux-Distribution gewählt werden. (Distribution bedeutet stark vereinfacht ausgedrückt: Ein auf Linux basierendes Betriebssystem, welches eine Reihe an Bedienelementen, graphischen Elementen und weiteren Programmen und Werkzeugen mit sich bringt.) Es gibt eine grosse Auswahl an nutzerfreundlichen Linux-Distributionen. Die Wahl fällt ohne tieferliegende Gründe auf Ubuntu, einfach weil es mir bereits etwas bekannt ist.

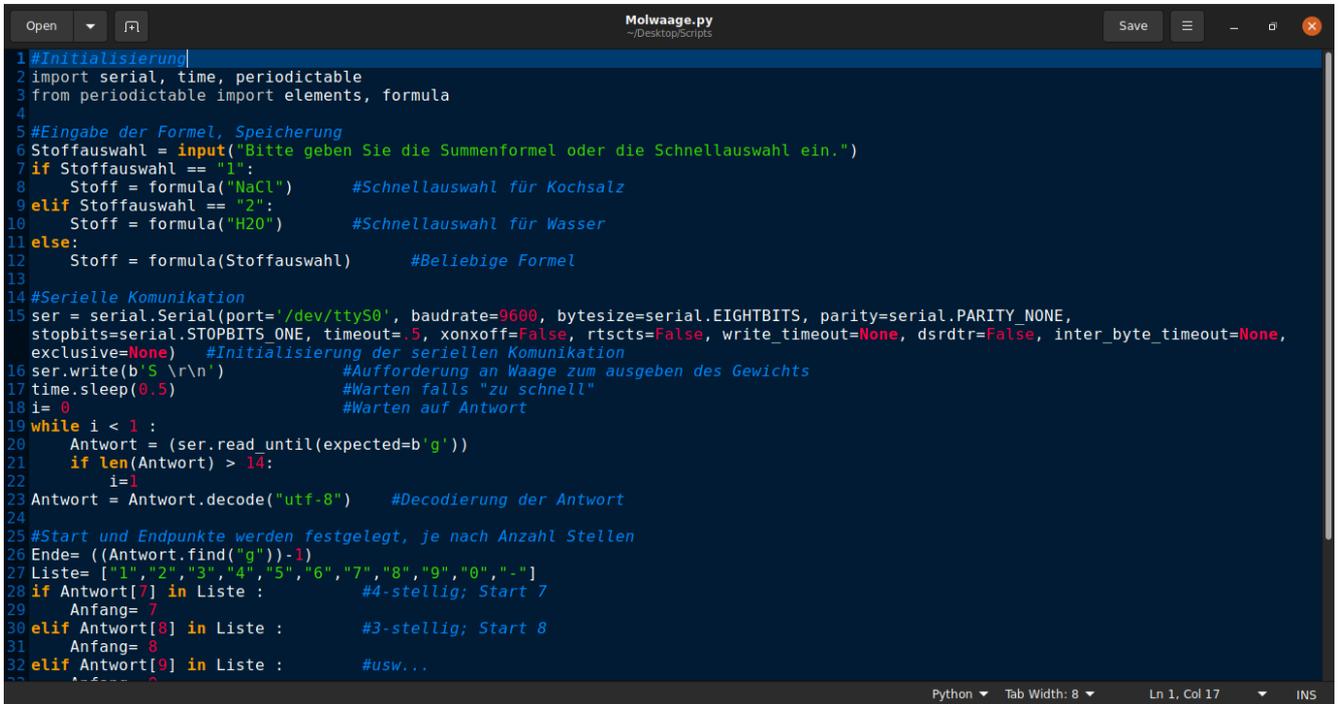
Um die Bedienung optisch attraktiv zu gestalten, wurde Ubuntu-Cinnamon gewählt, eine Version von Ubuntu mit dem vorinstallierten Grafikpaket Cinnamon.

3.2.5 Entwicklungsumgebung

Auf Linux stehen viele Entwicklungsumgebungen zur Verfügung. Eine mögliche Wahl wäre die Python-eigene Python-IDLE, Pythons integrierte Entwicklungs- und Lernumgebung. Der Fokus dieses Projekts liegt aber nicht auf dem Lernen, sondern auf der Entwicklung. Deshalb und wegen persönli-

² rs-online.com: <https://ch.rs-online.com/web/generalDisplay.html?id=ideen-und-tipps/rs232-kabel-leitfaden>, Stand: 11.12.2021

chen Empfehlungen sowie der grossen Beliebtheit unter vielen Usern wurde zu Atom gegriffen, einem Texteditor, der gleich wie Python-IDLE das automatische Einfärben von Funktionen, Variablen usw. ermöglicht.



```
1#Initialisierung
2import serial, time, periodictable
3from periodictable import elements, formula
4
5#Eingabe der Formel, Speicherung
6Stoffauswahl = input("Bitte geben Sie die Summenformel oder die Schnellauswahl ein.")
7if Stoffauswahl == "1":
8    Stoff = formula("NaCl") #Schnellauswahl für Kochsalz
9elif Stoffauswahl == "2":
10    Stoff = formula("H2O") #Schnellauswahl für Wasser
11else:
12    Stoff = formula(Stoffauswahl) #Beliebige Formel
13
14#Serielle Kommunikation
15ser = serial.Serial(port='/dev/ttyS0', baudrate=9600, bytesize=serial.EIGHTBITS, parity=serial.PARITY_NONE,
16stopbits=serial.STOPBITS_ONE, timeout=.5, xonxoff=False, rtscts=False, write_timeout=None, dsrdtr=False, inter_byte_timeout=None,
17exclusive=None) #Initialisierung der seriellen Kommunikation
18ser.write(b'S \r\n') #Aufforderung an Waage zum ausgeben des Gewichts
19time.sleep(0.5) #Warten falls "zu schnell"
20i = 0 #Warten auf Antwort
21while i < 1 :
22    Antwort = (ser.read_until(expected=b'g'))
23    if len(Antwort) > 14:
24        i=1
25Antwort = Antwort.decode("utf-8") #Decodierung der Antwort
26
27#Start und Endpunkte werden festgelegt, je nach Anzahl Stellen
28Ende = ((Antwort.find("g"))-1)
29Liste = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-"]
30if Antwort[7] in Liste : #4-stellig; Start 7
31    Anfang = 7
32elif Antwort[8] in Liste : #3-stellig; Start 8
33    Anfang = 8
34elif Antwort[9] in Liste : #usw...
```

Abbildung 2: Atom im Einsatz

Weiter unterstützt Atom automatische Vervollständigung, parallele Bearbeitung von mehreren Programmen gleichzeitig und ist personalisierbar. 18% der online befragten Stack-Overflow-Nutzerinnen und -Nutzer gaben an, Atom zu nutzen, womit es zu den Top 10 (Platz 9) der beliebtesten Code-Editoren gehört.³

3.2.6 Serieller Monitor

Um Experimente oder Tests zur seriellen Kommunikation mit der Waage machen zu können, braucht es einen seriellen Monitor. In diesem werden Einstellungen zur Verbindung gemacht und diese eröffnet, anschliessend können Befehle gesendet und einkommende Nachrichten betrachtet werden. Das ist wichtig, um Befehle zu testen und herauszufinden, was, wie und wie schnell die Waage auf solche antwortet.

Das gewählte Programm hierfür heisst CuteCom. Es ist unter Linux sehr einfach zu installieren. Schön an diesem sind die vielen Einstellungsmöglichkeiten und eine Übersicht über die letzten ein- und ausgegangenen Nachrichten.

3.2.7 Bibliotheken

Bibliotheken sind Sammlungen von Funktionen / Objekten für eine Programmiersprache, in diesem Falle Python, die von Drittpersonen oder dem Entwickler der Sprache erstellt wurden. Bei den hier benutzten handelt es sich um frei zugängliche Bibliotheken. Sie enthalten Programmbausteine, die verschiedenste Aufgaben erfüllen, die nicht neu programmiert werden müssen.

³ Gemäss software.com (Geoff Stevens): <https://www.software.com/src/ranking-the-top-5-code-editors-2019>, Stand: 27.12.2021

3.2.7.1 pySerial

PySerial ist eine Bibliothek für Python, die das Öffnen von seriellen Verbindungen in Python ermöglicht. Zum Eröffnen der Verbindungen gibt es viele Einstellungsmöglichkeiten, wie in CuteCom. Es gibt Funktionen zum Senden und Empfangen von Nachrichten.

3.2.7.2 Periodictable

Projektbeschreibung von Periodictable, übersetzt von der offiziellen Seite:⁴

Dieses Paket bietet ein Periodensystem der Elemente mit Unterstützung für Masse, Dichte und Röntgen- / Neutronenstreuungsinformationen. Massen, Dichten und natürliche Häufigkeit stammen vom NIST (National Institute of Standards and Technology) Physiklabor, sind aber nicht durch NIST-Wissenschaftler kritisch bewertet und geprüft worden.

Periodictable ermöglicht das Aufrufen von Atom- und Molekülmassen und wird genau dafür verwendet.

3.2.7.3 Time

Time ist eine Bibliothek, streng genommen ein Modul, das Zeit-basierte Funktionen ermöglicht. Dieses Modul muss zwar nicht speziell installiert aber trotzdem in der Programminitialisierung aufgerufen werden. Es ist in der Python 3.7-Installation integriert.

3.3 Entwicklungsvorgang

3.3.1 Allgemein

Der Entwicklungsvorgang besteht aus vier Aktivitäten: Recherche, Vorbereitung, Experimente und Programmierung.

Die Recherche ist am Anfang dieses Projekts der erste Schritt. Sie ermöglicht eine bessere Aufklärung über die Ausgangslage und mögliche Lösungsorientierungen. Sie ist im Weiteren beim Finden und Wählen der Hilfsmittel wichtig.

Mit der Vorbereitung werden die Grundlagen und die Umgebung für die Entwicklung geschaffen. Die nötigen Programme für die weiteren Schritte werden hier installiert und die technischen Geräte vorbereitet.

Die Experimente beinhalten verschiedene Versuche im Hard- oder Softwarebereich. Die Funktionsweise von einzelnen Teilen soll dadurch geprüft und verstanden werden.

Bei der Programmierung wird Programmcode geschrieben und bearbeitet.

In dieser Arbeit, wie auch in vielen anderen Projekten, fallen die oben beschriebenen Aktivitäten nahe zusammen. Für die Entstehung eines Programmabschnitts wird zwischen Recherche, Experimenten und Programmanpassungen hin- und hergewechselt oder diese werden parallel betrieben. Mit einem solchen Entwicklungsstil wird die Kenntnis laufend vertieft und es gibt eine gewisse Flexibilität, da laufend auf neu auftretende Probleme mit z.B. Recherche eingegangen werden kann.

3.3.2 Hilfen bei der Entwicklung

Beim Programmieren im Allgemeinen steckt man oft in scheinbar aussichtslosen Situationen. Gründe dafür, dass ein Programm nicht funktioniert, können alles von Rechtschreibbefehlern bis völlig falsch angewendete Funktionen sein.

⁴ pypi.org: <https://pypi.org/project/periodictable/>, Stand: 11.12.2021

Mit Zeit und Mühe und dem Überprüfen des gesamten Programms können solche Fehler behoben werden. Aber auch mit externer Hilfe kann in solchen Momenten ein Weg aus der Sackgasse gefunden werden. Es gibt Fehler, die dem Programmierenden selbst, nach langer Suche und Frustration, nicht auffallen. Unbeteiligte Betrachter sehen diese möglicherweise direkt.

Dies kann auch im Internet geschehen, dort sind viele Plattformen zu finden, auf denen Lösungen zu problematischen Programmen ausgetauscht werden. Auch wenn man diese Austauschplattformen nicht aktiv nutzt, also nicht die eigenen Fragen mit Hoffnung auf Hilfe dort platziert, können sie von grossem Nutzen sein. Viele Probleme wurden bereits von anderen veröffentlicht und beantwortet. Dadurch sind diese Seiten sehr wertvolle Quellen beim Programmieren und waren auch in dieser Arbeit eine wichtige Ressource.

Es gibt verschiedene solcher Webseiten, die hier konsultierte nennt sich Stack Overflow. Sie half beim Verstehen verschiedenster Funktionen, die im Rahmen der Entwicklung aufgetaucht sind, wobei nur bereits bestehende Fragen und Antworten konsultiert wurden.

3.4 Leistungen und Resultate

Phase	Meilenstein	Aktivität	Resultat
1	Installationen und Vorbereitung		
1.1	Betriebssystem und Konfiguration	Betriebssystem aufgesetzt und Berechtigungen erteilt	Betriebssystem einsatzfähig
1.2	Serieller Monitor, Programmiersoftware und Umgebung	CuteCom, Python3.7 und Atom installiert	Alle einsatzfähig
1.3	Vorbereitung der Waage	Verschiedene Einstellungen gemacht	Waage einsatzfähig
1.4	Installation von Bibliotheken	pySerial, Periodictable und Time installiert	Bibliotheken einsatzfähig
2	Entwicklung		
2.1	Test serieller Schnittstelle	Analyse von Ein- & Ausgabe	Feststellungen gemacht
2.2	Serielle Kommunikation mit Programm	Verbindungsaufbau, Senden von Befehlen und Empfangen von Antworten	Verbindung und Kommunikation geglückt
2.3	Beispielprogramme mit Periodictable	Aufruf von Elementen, Formeln und molaren Massen	Bibliothek verstanden
2.4	Testprogramm zur Integer-Konversion	Text kann in Zahl gewandelt werden	String/Integer-Problem gelöst
2.5	String-Manipulation	Herausholen des gemessenen Wertes aus der Antwort der Waage	String-Manipulation verstanden, Extrahieren des Wertes geglückt
2.6	Gesamtprogramm	Zusammenfügen der Teilprogramme und Fehlerbehebungen	Gesamtprogramm funktioniert

Tabelle 6: Leistungen und Resultate

Nicht aufgeführt sind Rechercheaufwand und Dokumentationsaufwand während der Entstehungsphasen. Die Phase 0, Recherche und Überlegungen, deren Kerngedanken und Erkenntnis in den Kapiteln 1 und 2 dargelegt wurden, ist ebenfalls nicht aufgeführt.

4 Lösungsbeschreibung

4.1 Einführung

Im Folgenden wird das Resultat der praktischen Arbeit chronologisch zur Entstehungsreihenfolge erläutert. Das heisst, die Programmteile werden nicht in ihrer Reihenfolge wie im finalen Programm, sondern in der Reihenfolge ihrer Entstehung geschildert. Das finale Programm wird nach den einzelnen Abschnitten zusammengefasst, die Bedienung kommentiert und die eingebauten Vorkehrungen für die Fehlerbehandlung erklärt. Das finale Programm ist ausserdem im Anhang (Kapitel 6.5) zu finden.

Wie im Kapitel 4.3 noch genauer beschrieben wird, sind für die Grundfunktion der Molwaage zwei Komponenten vorhanden: Das gemessene Gewicht der Probe und die molare Masse. Die Kapitel 4.2 und 4.4 erklären das Erlangen des Waageresultats, während dazwischen, im Kapitel 4.3 alles von der molaren Masse bis zur Ausrechnung und Ausgabe des Resultats geklärt wird. Diese Abfolge ist, wie oben genannt, durch die zeitliche Abfolge bei der Entwicklung begründet.

Die in diesem Kapitel vorkommenden Befehle und deren Erläuterungen stammen von einer Dokumentation von Mettler Toledo, beziehungsweise sind von ihr übersetzt oder inhaltlich übernommen worden.⁵

Um beispielsweise Funktionen zu erklären, wurden in diesem Kapitel Platzhalter-Variablen oder Werte eingesetzt, die aufzeigen, wo bei einer Funktion später ein Wert stehen wird. Diese Platzhalter sind in Grossbuchstaben und kursiv abgebildet, um eben darauf hinzuweisen. Weiter ist der Programmcode zur besseren Abgrenzung vom Normaltext kleiner geschrieben und eingerückt.

4.2 Kommunikation Waage – PC

Der Weg des Signales beginnt bei der Waage. So beginnt auch die Entwicklung bei der Waage und bei der Frage, wie denn das ermittelte Gewicht einer Probe zum Computer gelangt.

Zuerst werden an der Waage Einstellungen (in deren Menu, siehe unten) gemacht. Anschliessend wurde die Verbindung aufgebaut und es wurde detektiert, wie die Waage angesprochen werden muss und wie sie auf verschiedene Befehle reagiert oder antwortet. Dies beides geschah jeweils erst im seriellen Monitor und wurde dann in Python reproduziert.

All dies war dank Mettler Toledo ein grosses Stück einfacher als ursprünglich vermutet. Deren Modelle teilen sich nämlich grösstenteils eine Reihe an Befehlen und standardisierter Inhalte für die serielle Kommunikation. Diese sind in einer frei zugänglichen, sehr detaillierten Dokumentation (siehe Ressourcen bei der Entwicklung, Kapitel 6.4) von Mettler Toledo zu finden. Eine weitere Ressource während der Entwicklung war die Bedienungsanleitung der Waage (ebenda).

4.2.1 Einstellungen

Folgende Einstellungen wurden auf der Waage gemacht (mit deren Anzeige und Tasten der Waage wurde zwischen diesen internen Einstellungen navigiert und diese angepasst):

- Peripheriegerät: Host
- Baudrate: 9600

5

Mettler-Toledo:
https://www.mt.com/dam/product_organizations/laboratory_weighing/WEIGHING_SOLUTIONS/PRODUCTS/M-T-SICS/MANUALS/en/Excellence-SICS-BA-en-11780711D.pdf, Stand: 13.11.2021

Diese waren für eine erfolgreiche Kommunikation nötig. Das Peripheriegerät bestimmt für die Waage, welches Gerät am anderen Ende der Verbindung ist und wie sie agiert und reagiert, in diesem Falle ist es Host = Computer. Die Baudrate bestimmt die Anzahl Zeichen, die pro Sekunde übertragen werden. Dieser Wert muss auf der Waage und dem Computer gleich sein.

Folgende Einstellungen wurden im seriellen Monitor CuteCom gemacht, welcher zum Testen der Verbindung, der Befehle und Antworten dient und im vorherigen Kapitel beschrieben wurde:

```
Baudrate: 9600
Port: /dev/ttySO
```

In CuteCom wird das am Ende von jeder Zeile stehende CRLF automatisch interpretiert und nicht angezeigt. Es bedeutet Carriage Return und Line Feed und ist für das Beginnen neuer Zeilen zuständig. CRLF kann auch als (`\r\n`) angezeigt oder behandelt werden, später wird dies nochmal erwähnt (in Python).

Analog dazu sieht der Code zum Öffnen der seriellen Verbindung in Python folgendermassen aus:

```
serial.Serial(port='/dev/ttySO', baudrate=9600)
```

Im finalen Programm wurden einige Details ergänzt, die die Stabilität der Verbindung verbessern sollen. Sie wurden mit Versuchen und Beispielen von Stack-Overflow gefunden. Im Code in Kapitel 6.5 sind diese Ergänzungen enthalten.

Um diese Verbindung im Programm einfach aufrufen zu können wird sie wie folgt in einer Variabel festgehalten:

```
ser = serial.Serial(...)
```

Das Objekt `serial` und die Funktion `.Serial` sind Teile der Bibliothek `pySerial` für Python.

4.2.2 Nachricht auf Waage anzeigen

Mit dem Befehl:

```
D_“TEXT“
```

wird die Waage auf dem integrierten Display einen Text anzeigen. Mit den Antworten:

```
D_A, D_I oder D_L
```

wird der Status der Verarbeitung rückgemeldet. *A steht für eine korrekte Ausführung, *I bedeutet «verstanden, aber zurzeit nicht ausführbar», *L bedeutet «verstanden, aber wegen falschen Werten oder fehlendem Display nicht ausführbar».

In CuteCom wird der Befehl:

```
D_“Hallo_Welt“
```

mit der Antwort:

```
D_A
```

erfolgreich ausgeführt. Einzig die Länge der geschickten Nachricht kann zu Unleserlichkeit auf der Anzeige der Waage führen (wegen der begrenzten Displaygrösse wird je nach deren Länge nur der letzte Teil der Nachricht angezeigt).

In Python sieht die dementsprechende Programmzeile folgendermassen aus:

```
ser.write('D „Hallo Welt“ r\n\'.encode('utf-8'))
```

oder in kurzer Form:

```
ser.write(b'D „Hallo Welt“ r\n\')
```

Vorab wurde wie in 4.2.1 gezeigt die serielle Verbindung eröffnet. Die Funktion `.encode('utf-8')` oder das `b` vor dem Nachrichteninhalte kodieren die Nachricht, was so immer geschehen muss (wie die Dekodierung der empfangenen Antworten). Das früher erwähnte CRLF zum Zweck der Initiierung einer neuen Zeile ist hier als `r\n` in abgekürzter Form anzutreffen, es muss in jeder Nachricht enthalten sein. Auch jede Antwort oder Rückgabe der Waage beinhaltet das CRLF, dies wird aber in der folgenden Besprechung nicht jedes Mal erwähnt.

4.2.3 Gewichtsabfrage

Zur Gewichtsabfrage gibt es viele Möglichkeiten. Die detektierte Masse der Probe kann auf mehrere Arten angefordert werden. Doch die zwei Grundlegenden und in dieser Anwendung wichtigsten Befehle sind:

Zum Aufrufen eines stabilen Wertes (Wird eine neue Probe aufgelegt und gewogen, dauert es eine kurze Zeit, bis der Messwert stabil ist. Die Waage wartet bis der Messvorgang vollständig abgeschlossen ist und der Wert sich eependelt und nicht mehr verändert, dann schickt sie ihn):

S

und zum sofortigen Aufrufen eines aktuellen Wertes (der möglicherweise instabilen und abweichend ist, da die Waage den aktuellen Wert nimmt, ohne auf ein stabiles Messresultat zu warten):

SI

Beim Detektieren einer Stoffmenge in einer wie in Kapitel 1 vorgestellten Unterrichtssituation dürfte ein stabiler Wert eher die richtige Wahl sein. Dies bringt aber auch eine kleine Problematik mit sich, deren Lösung später noch gezeigt wird (Kap. 4.5.5).

Weiter sieht eine Antwort der Waage bei korrekter Ausführung folgendermassen aus:

```
S_S_GEMESSENERWERT_EINHEIT
```

Mögliche Antworten bei Fehlern sind:

S_I, S_+ oder S_-

wobei I bedeutet, die Waage ist mit einem anderen Vorgang beschäftigt und kann nicht ausführen, + und – bedeuten jeweils, die Probe ist über respektive unter dem messbaren Bereich. Der erste Buchstabe einer Antwort bedeutet immer die Art von Aktion, die gerade gemacht wird. S steht für Wägen und D für Display. Die jeweils zweiten Buchstaben oder Zeichen beschreiben den Ausführungsstatus und haben verschiedene Bedeutungen, die sich teils aus der englischen Bedeutung ableiten lassen.

In CuteCom funktioniert der Befehl problemlos. Die Antwort wird, sobald sie empfangen wurde, angezeigt:

```
S_S(...*)_0.00_g
```

Mit dem * in (...*) wird angedeutet, dass die genaue darin enthaltene Anzahl an Leerzeichen variiert. Dies wird in Kapitel 4.4 im Rahmen der String-Manipulation und Bearbeitung des Rückgabewertes zum Zweck seiner Verarbeitung genauer betrachtet.

In Python wird mit:

```
ser.write(b'S \r\n')
```

der Befehl gesendet (wieder mit dem b kodiert) und nach einer kurzen Pause wegen der Verarbeitungsdauer der Waage:

```
time.sleep(0.5)
```

```
ser.read()
```

deren Antwort gelesen, und folgendermassen abgespeichert:

```
Antwort = (ser.read())
```

Und um das weiter oben beschriebene CRLF (welches am Ende der Antwort, nach dem g steht) loszuwerden, kann aus dieser Antwort alles nach dem g gelöscht werden:

```
Antwort = (ser.read_until(expected=b'g'))
```

Es stellt sich heraus, dass eine Wartezeit von 0.5 Sekunden in manchen Fällen nicht reicht, und zwar dann, wenn die Waage noch einige weitere (Zehntel-) Sekunden zur Ausgabe des stabilen Wertes braucht. Anders gesagt ist der Wäge-Vorgang noch nicht abgeschlossen, und deshalb wird mit dem obigen Programm-Abschnitt eine leere Eingabe gemessen. Dies resultiert in Programmabbrüchen wegen Folgefehlern. Um diesen Fehler zu beheben, wird der Befehl so lange wiederholt, bis der Rückgabewert eine bestimmte Anzahl Zeichen erreicht. Da diese Vorkehrung eine Funktion der String-Manipulation enthält, wird sie erst im Kapitel 4.5.1 vorgestellt.

4.3 Mol-Rechnung

Um die inhaltliche Korrektheit folgender Erläuterungen zu garantieren, wurden mehrere Artikel des Online-Lexikons Chemie.de konsultiert und Inhalte übernommen.⁶

Die Stoffmenge n kann folgendermassen errechnet werden (die wichtigste Formel der folgenden Erklärungen):

$$n = \frac{m}{M}$$

wobei m der Masse der Probe und M der molaren Masse entspricht. Das Herausfinden von n ist das grosse Ziel der nächsten Schritte. Die Masse m wird von der Waage geliefert, M wird mit Hilfe der Bibliothek Periodictable gefunden, aber nicht direkt.

Die Einheit für die Stoffmenge n ist Mol. Sie und die (absolute) Teilchenzahl N stehen direkt proportional zueinander. Dabei gilt:

$$N = n \cdot N_A$$

wobei N_A die Avogadro-Konstante bedeutet.

Die Einheit der Masse m wird Gramm sein. Die molare Masse M ist folgendermassen bestimmt:

$$M = \frac{m}{n} = N_A \cdot m_M$$

Die Molekülmasse m_M in der Einheit u hat den gleichen Zahlenwert wie die molare Masse M . Die Molekülmasse m_M kann mit Periodictable aufgerufen werden, somit sind sowohl m als auch M der ganz obigen Formel erreichbar, und es muss keine weitere Umrechnung gemacht werden.

⁶ chemie.de: <https://www.chemie.de/lexikon/> (mehrere Artikel), Stand (alle): 13.11.2021

Der Aufruf von M ist Objekt dieses Unterkapitels 4.3, während das Erhalten des Wertes für m in 4.4 geklärt wird.

4.3.1 Aufruf einer molaren Masse

Um nicht eine eigene Datenbank an molaren Massen anlegen zu müssen, wird auf die Bibliothek `Periodictable` zugegriffen. Diese dreht sich wie der Name schon sagt rund um die Thematik Periodensystem und macht chemische Rechnungen oder ähnliches in Python möglich.

Wie in den vorangehenden Erklärungen erwähnt, sind die numerischen Werte für die molare Masse und die Molekülmasse gleich. Da in Python rein in Zahlen und ohne Einheit gerechnet wird, kann problemlos anstelle der molaren Masse die Molekülmasse aufgerufen und zum Verrechnen verwendet werden. Denn die Molekülmasse ist mit:

```
BEISPIELSTOFF.mass
```

einfach aufgerufen (Achtung: `.mass` heisst Molekülmasse und hat nichts mit der Masse M der Probe zu tun). An der Stelle von `BEISPIELSTOFF` steht ein Element oder eine Stoffverbindung. Handelt es sich um eine Stoffverbindung, so muss diese als Formel definiert sein.

Um eine Formel zu definieren, wird die Funktion `formula` verwendet. Diese nimmt einen String, also eine Zeichenfolge, und ruft die dementsprechende Formel auf. Die Gross- und Kleinschreibung ist Chemie-standardmässig, also wird immer der erste Buchstabe des Elementkürzels gross und der Rest klein geschrieben. Wie oft ein Element (oder Atom) im Molekül vorkommt wird wie im normalen chemischen Umgang hinter dem Elementkürzel geschrieben, aber nicht hoch- oder tiefgestellt. Folgendes Beispiel für Wasser:

```
formula(„H2O“)
```

Würde die molare Masse oder die Molekülmasse einer Verbindung von Hand berechnet, so multiplizierte man die einzelnen Elements-Atommassen mit ihrer Anzahl Vorkommnis im Molekül und summierte diese. Dies bleibt hier erspart und wird von der Funktion selbst ausgeführt. Wie im Schlusswort noch erwähnt wird, gehen die Möglichkeiten von `Periodictable` weit über diese einfache Anwendung hinaus. Hier werden nur einfache Stoffe, keine Lösungen oder sonstiges, vorgesehen.

4.3.2 Kurzbefehle

Um schnell und ohne Formeleingabe auf der Tastatur einen Stoff aus einer kleinen Liste aufrufen zu können, werden Kurzbefehle verwendet. Diese könnten unter Umständen auch mit der Taste auf der Waage ausgewählt werden. In dem Falle müsste eine Funktion programmiert werden, welche detektiert, wie oft die Taste gedrückt wird.

Aber dies bringt wieder viele Komplikationen mit sich, eine benutzerfreundliche und zufriedenstellende Programmierung wäre nicht ganz unproblematisch. Denn die Zeit, in der Tastendrücke erwartet werden oder die zwischen ebendiesen liegt, müsste festgelegt werden. Zwar wäre das Verwenden der Taste sehr schön, birgt aber erheblichen Aufwand.

Viel einfacher ist es, die Kurzbefehle mit der Tastatur einzugeben. Im finalen Programm, welches in Kapitel 6.5 ersichtlich ist, stehen 5 im Labor häufig eingesetzte Stoffe zur Schnellauswahl bereit.

Mit:

```
input(„Bitte Kurzbefehl eingeben“)
```

wird die Eingabe gemacht respektive ermittelt. Im Kommandozeilenfenster, indem das Programm ausgeführt wird, wird die Nachricht angezeigt und eine Zahl kann eingegeben werden. Mit:

```
Stoffauswahl = input(„...“)
```

wird die Eingabe gespeichert und mit:

```
if Stoffauswahl == „1“:
```

```

Stoff = formula(„NaCl“)
elif Stoffauswahl == "2":
    Stoff = formula("H2O")
elif Stoffauswahl == "3":
    ...

```

kann die Eingabe mit einer Reihe an Beispielen verglichen werden. Die Bedeutung von if & elif entspricht in etwa falls & sonst falls. In anderen Worten: Ist es 1? Wenn nicht, ist es 2? Und so weiter. Die Funktion

```
Stoff = formula(„...“)
```

Speichert die Formel für die spätere Umrechnung, wie sie in 4.3.4 erläutert wird.

4.3.3 Formeleingabe

Noch einfacher und zugleich genauere Eingaben erlaubend ist das Nutzen der Tastatur zum Eingeben von ganzen Formeln. Wieder kann mit:

```
input(„Bitte geben sie eine Formel ein“)
```

eine Eingabe gemacht respektive verarbeitet werden. Sie wird aber direkter verarbeitet:

```
Stoffauswahl = input(„...“)
Stoff = formula(Stoffauswahl)
```

Diese Funktion wird mit der oben beschriebenen Kurzbeleg-Erkennungsfunktion aneinandergelängt. Wird mit if & elif kein Kurzbeleg / keine Schnellauswahl erkannt, so wird mit else die Stoffauswahl direkt als Formel gespeichert.

4.3.4 Stoffmengenberechnung

Zuerst wird die molare Masse des gerade festgelegten Stoffes aufgerufen (wie in 4.3.1 beschrieben):

```
molare_Masse = Stoff.mass
```

und dann die Stoffmenge errechnet (wie in der Einführung erklärt zu 4.3 erklärt):

```
Stoffmenge = Masse/molare_Masse
```

Nachdem nun das Herausfinden des Stoffes durch eine Eingabe des Nutzers und die darauffolgende Detektierung der molaren Masse geklärt sind, folgt nun in 4.4 das Erhalten des Wertes für die Masse.

4.4 Bearbeitung des Rückgabewertes / String-Manipulation

Der Rückgabewert der Waage muss auf verschiedene Arten bearbeitet werden, um vom Programm zur Errechnung der Stoffmenge gehandhabt werden zu können. Nämlich dürfen nur noch die Ziffern und das Komma (wird als Punkt angezeigt) ohne jegliche Leerzeichen oder Einheit stehen. Weiter muss es als Zahlenwert (indem Falle als Float; also als Zahl mit Nachkommastellen) definiert sein, und nicht als Zeichenfolge (das wäre ein String). Die Bearbeitung erfolgt in mehreren Schritten, alle werden in diesem Unterkapitel erklärt.

4.4.1 Decodierung

Die Rückgabewerte der Waage sind kodiert in utf-8. Es wurde bereits erklärt, dass alle Ein- und Ausgaben kodiert sind, beziehungsweise kodiert und dekodiert werden müssen.

In Python wird der bereits in einer Variabel gespeicherte Wert in derselben in decodiert gespeichert. Die Decodierung einer Variabel geschieht mit:

`VARIABLE.decode(ART DER CODIERUNG)`

Also in diesem Falle:

`Antwort = Antwort.decode(„utf-8“)`

Die Variable Antwort ist der gespeicherte Rückgabewert der Waage. Sie wird decodiert und wieder unter gleichem Namen gespeichert.

4.4.2 String-Manipulation 1: Endpunkt

Die String-Manipulation (oder deutsch Zeichenfolgen-Manipulation oder Bearbeitung) ist eine Reihe in Python selbst enthaltener Funktionen, die komplexe Bearbeitungen von Text zulassen. Mit ihr kann der Ort eines Zeichens im Text, das Zeichen an einem Ort im Text oder der Textabschnitt zwischen zwei Orten im Text aufgerufen werden, um nur einige zentrale und hier vorkommende Beispiele zu nennen.

Um den gewünschten Teil, also die Zahlen ohne Leerzeichen und Einheit usw., zu extrahieren, müssen der Anfang und Schluss des gewünschten Abschnitts bekannt sein. In verschiedenen Tests wurde probiert, gleichbleibende Eigenschaften festzuhalten. Es hat sich gezeigt, dass dies nicht ganz einfach ist. Je nach Grösse der Zahl (im Rückgabewert) verändert sich der Anfangspunkt.

Der Text sei in einer Variabel gespeichert, so wird wie folgt ein Zeichen aufgerufen:

`VARIABLE.find(„ZEICHEN“)`

Der Endpunkt der gesuchten Zahl liegt vor der Einheit, also vor dem g. Die Position dieses wird auffindig gemacht:

`Antwort.find(„g“)`

Die Definition des Endpunktes als:

`Ende = ((Antwort.find(„g“))-1)`

erwies sich als korrekt, das vermeintliche Leerzeichen zwischen Zahl und Einheit (Gramm; g) wurde dank dem -1 mitausgeschlossen. In der Variabel Ende wurde dieser Endpunkt sogleich gespeichert.

4.4.3 String-Manipulation 2: Startpunkt

Der Startpunkt der Zahl ist noch ein Stück schwieriger zu finden als der Endpunkt. Wie bereits erwähnt, variiert er je nach Anzahl Stellen der Zahl. Immerhin konnte festgestellt werden, dass er sich in jedem Fall zwischen den Stellen sieben bis zehn befindet.

Die erste Stelle von links, die kein Leerzeichen enthält, sei der Startpunkt. Ausser den Ziffern null bis zehn könnte es sich bei der ersten Stelle auch um ein Minuszeichen handeln. In einer Liste werden alle Möglichkeiten erfasst, um nachher einfach abgerufen und verglichen werden zu können.

`Liste = [„1“, „3“, „4“, „5“, „6“, „7“, „8“, „9“, „0“, „-“]`

Mit:

`TEXT[STELLE]`

wird das Zeichen an der gewünschten Stelle eines Textes aufgerufen. Es kann mit in herausgefunden werden, ob an einer Stelle (X) eines der zur Frage stehenden Zeichen vorkommt:

`Antwort[X] in Liste`

Mit der if (falls) Funktion wird eine Aktion ausgeführt, insofern die gegebene Anforderung erfüllt ist. So wird der Startpunkt der Zahl als siebte Stelle im Text definiert, wenn sich an ihr eines der ausschlaggebenden Zeichen befindet:

```
if Antwort[7] in Liste:  
    Anfang = 7
```

Mit elif (Abkürzend für else if; sonst falls) wird die Funktion bei einer Nichterfüllung mit einer weiteren Anforderung und einer weiteren möglichen Aktion wiederholt. Werden diese Funktionen aneinandergereiht, kann in der gewünschten Reihenfolge eine Reihe an Begebenheiten geprüft und eine dementsprechende Aktion ausgeführt werden (hier das Festsetzen des Startpunkts, je nachdem, welches die erste Stelle von links mit einem Zeichen ist). Wie beschrieben müssen nur die Stellen sieben bis zehn getestet werden, vorher steht in keinem Fall etwas anderes als ein Leerzeichen. Die Reihenfolge ist von links nach rechts, also in aufsteigenden Zahlen. Das Programm sieht folgendermassen aus:

```
if Antwort[7] in Liste:  
    Anfang = 7  
elif Antwort[8] in Liste:  
    Anfang = 8  
elif Antwort[9] in Liste:  
    Anfang = 9  
elif Antwort[10] in Liste:  
    Anfang = 10
```

4.4.4 String-Manipulation 3: Aufruf eines Abschnitts

Die Funktion:

```
TEXT[STARTPUNKT:ENDPUNKT]
```

ruft den Textabschnitt vom Start- bis Endpunkt auf. Diese sind (wie in 4.4.2 und 4.4.3 geklärt) als Variable gespeichert und können aufgerufen werden. Der Textabschnitt mit der Zahl kann folgendermassen aufgerufen werden:

```
Antwort[Anfang:Ende]
```

Die Zahl ist bisher als Zeichenfolge definiert oder vorhanden. Damit ein Wert von Python für Operationen verwendet werden kann, muss er von der Art her ein numerischer Datentyp sein (ganze Zahl, Gleitkommazahl oder komplexe Zahl). Glücklicherweise kann in Python eine Reihe an Ziffern-Zeichen zu einer „echten“ Zahl gewandelt werden:

```
float(ZAHL)
```

Diese Funktion kann eine Zahl, die als Zeichenfolge (String) definiert ist, als effektiv fürs Programm verwendbare Zahl mit Nachkommastellen umdefinieren. Diese wird in diesem Projekt dann gleich in einer neuen Variablen gespeichert. So ergeben die zwei hier erklärten Funktionen die Programmzeile:

```
Masse = float(Antwort[Anfang:Ende])
```

Somit ist die nötige Form erreicht.

4.5 Übersicht

Die Kapitel 4.2, 4.3 und 4.4 lösten alle unabhängig voneinander Teilprobleme, die nun im Zusammenspiel das entstandene Programm ausmachen.

4.5.1 Initialisierung: Aufruf der Bibliotheken

Im Programm werden Funktionen aus, teils externen, Bibliotheken verwendet. Diese müssen Anfangs des Programms aufgerufen werden. Mit:

```
import serial, time, periodictable
```

wird dies getan. Zwei speziell aufzurufende Funktionen der Bibliothek Periodictable werden mit:

```
from periodictable import elements, formula
```

bereitgestellt.

4.5.2 Anzeigen der Masse

Es könnte praktisch sein, parallel zur errechneten Stoffmenge auch die Masse am Computerdisplay einsehen zu können. Diese Funktion ist einfach zu programmieren. Allerdings muss die Zahl als Zeichenfolge aufgerufen werden, was mit der Funktion `str()` (ähnlich wie bei der float-Wandlung in 4.4.4) geschieht:

```
print(str(Masse), „ g“)
```

4.5.3 Anzeigen der Stoffmenge

Das Detektieren der Masse und der molaren Masse wurde behandelt und die Resultate werden hier für die finale Berechnung der Stoffmenge aufgegriffen.

```
Stoffmenge = float(Masse)/molare_Masse
```

Da die Resultate viele Nachkommastellen enthalten, bietet es sich an, zu runden. Die Funktion:

```
round(ZAHL,ANZAHL_DEZIMALSTELLEN)
```

rundet eine Zahl auf die gewünschte Menge an Nachkommastellen. Für die Zwecke im Unterricht muss das Resultat nicht in hoher Genauigkeit vorliegen. Mit drei Dezimalstellen sollte das Resultat zugleich übersichtlich und aussagekräftig genug sein:

```
print(round(Stoffmenge,3), „ Mol“)
```

und wird hiermit inklusive Masseinheit im Kommandozeilenfenster ausgegeben.

4.5.4 Vorkehrungen

Wie an manchen Stellen erwähnt, scheint eine Wartezeit von 0.5 Sekunden zwischen der Aufforderung zum Ausgeben eines Wertes in manchen Fällen nicht zu genügen. Am einfachsten wäre es, den Befehl an die Waage zum Senden eines stabilen Wertes durch den Befehl zum Senden eines sofortigen Wertes zu ersetzen (siehe 4.2.3). Ein stabiler Wert ist doch aber vorzuziehen.

Es gibt auch einen Weg, das Problem ohne Ausweichen auf den instabilen Wert zu lösen. So kann einfach das Lesen des Rückgabewertes der Waage wiederholt werden, bis dieser eine Anzahl Zeichen erreicht, die nur bei einer geglückten Messung erreicht wird. Die beobachtete Mindestzeichenlänge der Antworten der Waage beträgt 15, also wird jede einkommende Antwort darauf geprüft, ob sie mehr als 14 Zeichen lang ist.

Mit:

```
len(Antwort)
```

wird diese Zeichenlänge aufgerufen und mit der Schleife:

```
while i < 1 :  
    Antwort = (ser.read_until(expected=b'g'))  
    if len(Antwort) > 14:  
        i=1
```

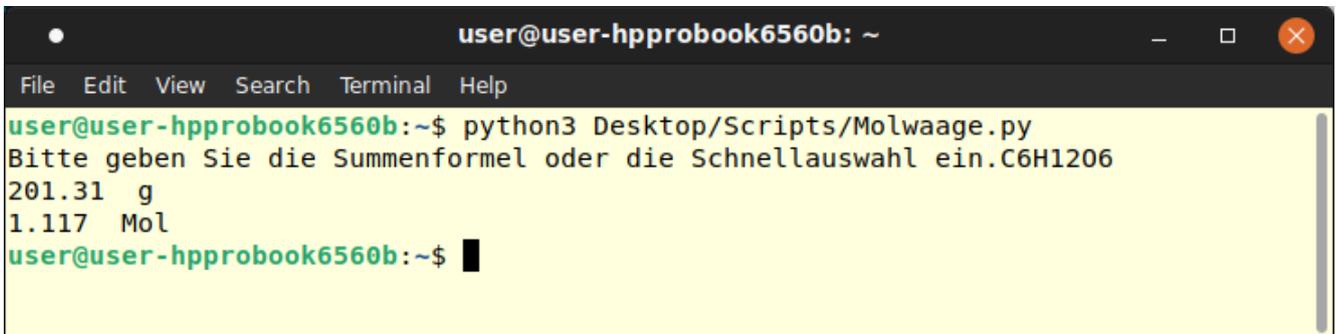
das erwähnte Messen des Rückgabewertes bis zum Erhalt eines zufriedenstellenden Wertes wiederholt. Dies verhindert Programmfehl Ausführungen, denn das Programm würde ohne diese Vorkehrung bei einer zu spät eintreffenden Antwort der Waage auf unlösbare Rechnungen stossen und terminieren, also abbrechen und eine Fehlermeldung ausgeben.

4.5.5 Anzeige

Das Programm sieht im Kommandozeilenfenster beispielsweise so aus:

```
Bitte geben Sie eine Summenformel oder einen Kurzbefehl ein. H2O  
18 g  
1 Mol
```

Und hier noch als Screenshot des Terminals (Kommandozeilenfenster auf Linux), wo das Programm aufgerufen und ausgeführt wird:



```
user@user-hpprobook6560b: ~  
File Edit View Search Terminal Help  
user@user-hpprobook6560b:~$ python3 Desktop/Scripts/Molwaage.py  
Bitte geben Sie die Summenformel oder die Schnellauswahl ein.C6H12O6  
201.31 g  
1.117 Mol  
user@user-hpprobook6560b:~$ █
```

Abbildung 3: Programm im Terminal

5 Nachüberlegungen

Nachdem die Motivation, die Entstehung und das technische Endprodukt vorgestellt wurden, möchte ich dieses letzte Kapitel einigen Interpretationen, Bewertungen und weiterführenden Ideen widmen. Zuerst werde ich das Endprodukt auf technischer Ebene bewerten und kommentieren. Anschliessend werde ich einige Gedanken zum Entstehungsprozess im Rückblick wie auch mögliche Weiterentwicklungen über diese Arbeit hinaus darlegen. Danach folgen Kommentare zur Unterrichts-anwendung, was verbessert werden könnte und welches weitere Potential noch offen liegt.

5.1 Endprodukt

Zum Einstieg in diese Nachbesprechung sollen die in Kapitel 2 behandelten Muss- und Kann-Anforderungen in Erinnerungen gerufen werden, um dann deren Erfüllung zu beurteilen:

Muss-Anforderung 1: Die Auswahl von mindestens drei Stoffen kann mit der Molwaage gewogen und automatisch umgerechnet werden → erfüllt. Das Programm zeigte seine Funktionsfähigkeit in mehreren Tests. Es unterstützt weit mehr als drei Stoffmöglichkeiten.

Praktisch steht das ganze Periodensystem zur Verfügung, und kann in einfachen Summenformeln verwendet werden (Kann-Anforderung Nr. 2 und 3 sind erfüllt). Allerdings wurden im Rahmen dieser Arbeit noch einige Gebiete ausgelassen, so etwa gelöste Stoffe und Isotope. Doch die Möglichkeiten der angewendeten Bibliothek reichen sehr weit über die gemachte Implementierung hinaus. Der Frage, was mit der Bibliothek Periodictable sonst noch angestellt werden kann, könnte man eine ganze Arbeit widmen.

Muss-Anforderung 2: Mindestens eine Art der visuellen Ausgabe des ermittelten Wertes ist vorhanden → erfüllt. Mit der Kommandozeile des Computers per Anzeige auf dessen Display (oder Beamer) können ermittelte Resultate visuell angezeigt werden. (Also eine ungeschönte und einfache Anzeige des Resultates, wie in Kann-Anforderung Nr. 4 beschrieben, die damit erfüllt ist.)

Der Prototyp wurde mit verschiedenen Stoffen und Proben getestet, wobei, nach Anpassungen (siehe 4.5.4) weder Abstürze, falsche Resultate oder Ähnliches beobachtet wurden. (Damit wird die Kann-Anforderung Nr. 6 erfüllt.)

Der Prototyp funktioniert. Doch er ist nichtsdestotrotz sehr eingeschränkt, was in der jetzigen Form den Einsatz im Unterricht nicht einfach ermöglicht.

Das Hauptproblem dabei ist die grafische Darstellung. Im Kommandozeilenfenster ist das Resultat schlecht lesbar. Um im Unterricht für die S.u.S. lesbar zu sein, müsste das Resultat grösser und auch zentraler im Bildschirm dargestellt sein. (Kann-Anforderung Nr. 5 ist nicht erfüllt.)

5.2 Weiterentwicklung

Im Folgenden werde ich auf die möglichen und nötigen Weiterentwicklungen eingehen, die an der Molwaage für einen komfortablen Einsatz im Unterricht nötig wären.

Das Wichtigste dabei wäre sicher eine schönere Darstellung (siehe auch Kann-Anforderung Nr. 5). Das Stichwort hierbei ist GUI, Graphical User Interface. Damit würden die Eingabe des Stoffes und die Ausgabe der errechneten Werte nicht in der Kommandozeile, sondern in einem graphischen Fenster geschehen, wo auch Bilder und Grafiken zum Einsatz kommen können.

Das Implementieren eines GUIs wäre mit absehbarem Aufwand machbar. Für Python gibt es viele vorgefertigte Programmbausteine und Bibliotheken. Es ist wahrscheinlich ein GUI zu finden, das nur noch minimale Abänderungen im Code benötigt und einfach die Kommunikation aus der Kommandozeile gross darstellt. Es könnten aber wieder unvorhergesehene Herausforderungen auftauchen. Vielleicht werde ich mich nach der Abgabe dieser Arbeit noch daran versuchen.

Kann-Anforderungen Nr. 7 und 8 drehen sich um das Betreiben der Molwaage ausserhalb des zur Entwicklung verwendeten Computers. Der Betrieb auf den Laptops der Lehrpersonen oder anderen Schulcomputern bring folgende Probleme mit sich.

Erstens verfügen diese in den meisten Fällen über keine serielle Schnittstelle. Dem könnte mit Adaptern Abhilfe geschaffen werden. Bei Adaptern ist die Mühsal der Treiberinstallation und andere Unbequemlichkeiten bekannt. Es müsste zur Auswahl des richtigen Adapters eine Recherche angestellt werden.

Zweitens läuft das Programm bisher nur auf Linux, mit manuell installierten Bibliotheken und mit installiertem Python 3.7. Sollte das Programm auf anderen Computern betreibbar sein, müssten dort Python 3.7 und alle verwendeten Bibliotheken installiert werden. Zusätzliche Komplikationen könnte das Betriebssystem auslösen, auf Windows und iOS könnten aus anderen unbekanntem Gründen Fehler auftreten.

Allenfalls wäre es möglich, das Programm ohne Installation zu betreiben. Bei Windows müsste das Programm dafür in der Form einer Anwendung, Executable, gebracht werden. Die Umwandlung von Python zu Anwendung (=Executable: exe Dateityp) ist auf verschiedene Arten möglich. Es gibt Wandler, die diese vollführen. Doch es ist mir bisher nicht klar, wie die Bibliotheken bereitgestellt werden. Bestimmt gäbe es dafür aber eine Lösung, vermutlich mit DLL-Dateien. DLL ist ein Dateityp für Programmbibliotheken, die von einer Anwendung verwendet werden. Auf Mac und iOS jedoch müsste ich noch weiter forschen.

Der Einsatz der Taste auf der Waage wäre wünschenswert gewesen (Kann-Anforderung Nr. 9). Es gäbe viele mögliche Funktionen, die damit belegt werden könnten. In der Entstehung dieser Arbeit hat sich diese Implementierung nicht ergeben. Das Programm könnte im Nachhinein dafür modifiziert werden, wenn sich eine passende Funktion zeigt. In der Dokumentation von Mettler Toledo wird die Implementation technisch erklärt, die nötigen Einstellungen oder Befehle für die Verwendung der Taste gezeigt.

5.3 Entstehung

Es mag auffallen, dass die Muss-Anforderungen sehr offen formuliert waren. Sie setzen zwar eine Zielmarke für die Entwicklung des Prototyps, aber sie lassen gleichzeitig viel Raum für Interpretation. Dies ist meiner Meinung nach auch nötig, denn die Kenntnis über viele technische Hürden fehlte vor dem Start der Arbeit noch.

Man könnte fragen, ob die Ziele zu offen formuliert waren, ob mit engeren Vorgaben die Arbeit hätte weiter geführt werden können. Im Rahmen dieser Arbeit ist es geglückt, die Muss-Anforderungen zu bewältigen, aus ihnen brauchbare und realistische Ziele zu machen. Nicht in allen Projekten oder Arbeiten ist es passend, wenn die Planung wie hier mit der Realisierung zusammenfliesst. Doch gerade wegen der grossen Flexibilität halte ich es im Nachhinein für eine effektive und die richtige Herangehensweise für diese Arbeit.

Die aufgestellten Kann-Anforderungen halte ich ebenfalls für gelungen. Ein Teil wurde erreicht, ein Teil nicht, so war das geplant. Die Unterteilung von Muss- und Kann-Anforderungen hat beim Fokus auf das Wichtigste geholfen, während die Sammlung an Kann-Anforderungen für die Weiterarbeit nach den erreichten Muss-Anforderungen einige Orientierungspunkte darstellte.

5.4 Weitere Gedanken

Wie bereits erwähnt ist die Bibliothek Periodictable sehr viel umfangreicher als die Implementierung in dieser Arbeit. So sehe ich in ihr einiges an Potential in Richtung Didaktik im Bereich Chemie. Nur ist es nicht möglich, im regulären Chemieunterricht mit Python zu arbeiten. Ich denke eher daran, dass im Fach Informatik, in welchem Python ohnehin thematisiert wird, auf Periodictable eingegangen werden könnte. Wie und in welchem Rahmen, dem könnte man eine ganze Arbeit widmen. Ich möchte

hiermit nur den Anreiz schaffen, dies einmal zu bedenken. Denn Fächer oder Disziplinen-übergreifende Aktionen können den Horizont auf ganz besondere Weise erweitern.

5.5 Schlusswort

Das Programmieren mit Python bereitete mir stets viel Freude. Gerade wegen der besonderen Mischung aus Leistungsfähigkeit und Erreichbarkeit gefällt mir der Umgang mit Python besonders. Ich habe in diesem Projekt einige neue Erfahrungen in Python gesammelt. Am spannendsten fand ich die String-Manipulation, welche mir komplett neu war und unter der ich mir im Vorhinein wenig vorstellen konnte.

Ganz nebenbei habe ich meine Fertigkeiten im Umgang mit Linux vertieft. Die Entwicklung mit Linux als Betriebssystem birgt effiziente Arbeitsabläufe und ist überraschend angenehm.

Im Rahmen dieser Arbeit habe ich nicht nur meine Programmierfertigkeiten trainiert und mein Wissen in verschiedenen technischen Bereichen vertieft, sondern auch Erfahrungen im Herangehen und Durchführen eines Projekts mit gewissen Vorgaben und Zielen gesammelt. Der ganze Weg, von der Idee zum Prototyp, war von unsicheren und aussichtslosen Momenten durchzogen. Es mussten anfangs viele Entscheidungen und Überlegungen getätigt werden und später waren Ausdauer und Kreativität gefragt. Auch wenn der entstandene Prototyp einfach sein mag und ich mir noch viele weitere Funktionen hätte erhoffen können, bin ich froh, den Kern der Herausforderung gemeistert zu haben.

Herrn Steiger danke ich sehr für die Betreuung bei dieser Arbeit und die wertvollen Tipps. Ich kann auf eine leerreiche und spannende Arbeit zurückschauen.

6 Anhang

6.1 Abkürzungsverzeichnis

Abkürzung	Erklärung
CRLF	Carriage Return / Line Feed
MA	Maturaarbeit
n.a.	Not available; Nicht verfügbar
usw.	und so weiter
z.B.	zum Beispiel
S.u.S.	Schülerinnen und Schüler

6.2 Bibliografie

Autor oder Herausgeber	Titel	URL
Chemie.de	Mol	https://www.chemie.de/lexikon/Mol.html (13.11.2021)
	Molare Masse	https://www.chemie.de/lexikon/Molare_Masse.html (13.11.2021)
	Molekülmasse	https://www.chemie.de/lexikon/Molek%C3%BClmasse.html (13.11.2021)
	Stoffmenge	https://www.chemie.de/lexikon/Stoffmenge.html (13.11.2021)
	Teilchenzahl	https://www.chemie.de/lexikon/Teilchenzahl.html (13.11.2021)
Mettler-Toledo	Standard Interface Command Set for Excellence Balances	https://www.mt.com/dam/product_organizations/laboratory_weighing/WEIGHING_SOLUTIONS/PRODUCTS/MT-SICS/MANUALS/en/Excellence-SICS-BA-en-11780711D.pdf (13.11.2021)
Pypi.org	Periodictable 1.6.0 – Project description	https://pypi.org/project/periodictable/ (11.12.2021)
Python.org (Python Software Foundation)	What is Python? Executive Summary	https://www.python.org/doc/essays/blurb/ (11.12.2021)
RS-Online.com	Alles, was Sie über RS232-Kabel wissen müssen	https://ch.rs-online.com/web/generalDisplay.html?id=ideen-und-tipps/rs232-kabel-leitfaden (11.12.2021)

Autor oder Herausgeber	Titel	URL
Stevens, Geoff/ Software.com	Code Time Data: Ranking the Top 5 Code Editors in 2019	https://www.software.com/src/ranking-the-top-5-code-editors-2019 (27.12.2012)

6.3 Abbildungsverzeichnis

Die 3 Abbildungen sowie das Bild auf dem Titelblatt sind persönlich aufgenommene Screenshots beziehungsweise Fotos.

6.4 Ressourcen bei der Entwicklung

Beim Folgenden handelt es sich nicht um zitierte, paraphrasierte oder anders in den Text eingeflossene Quellen, sondern um in der praktischen Arbeit konsultierte Ressourcen, die, wie im Text erwähnt, bei der Entwicklung hilfreich waren:

Titel oder Bezeichnung	URL
MT EL3002 MANUAL	https://www.mt.com/dam/mt_ext_files/Editorial/Generic/7/EducationLine_Editorial-Generic_1149086127154_files/el-ba-d-11780853b.pdf
Reference Manual: MT-SICS Interface Commands	https://www.mt.com/dam/P5/labtec/17_Miscellaneous/RM_Advanced_and_Standard_Level_Balances_SICS_EN.pdf
Stack Overflow	https://stackoverflow.com/

6.5 Programmcode

Der folgende finale Programmcode wurde zur erleichterten und authentischen Leserlichkeit Python-typisch eingefärbt (die roten Stellen mit den # sind Kommentare und nicht Teil des Programms). Auf einen Screenshot wurde aus Gründen der Leserlichkeit verzichtet.

```
#Initialisierung, Import (laden) der Bibliotheken / Module
import serial, time, periodictable
from periodictable import elements, formula

#Eingabe der Formel durch User*in, Speicherung
Stoffauswahl = input("Bitte geben Sie die Summenformel oder die Schnellauswahl ein.")
if Stoffauswahl == "1":
    Stoff = formula("NaCl") #Schnellauswahl für Kochsalz
elif Stoffauswahl == "2":
    Stoff = formula("H2O") #Schnellauswahl für Wasser
elif Stoffauswahl == "3":
    Stoff = formula("C6H12O6") #Schnellauswahl für Zucker
```

```

elif Stoffauswahl == "4":
    Stoff = formula("C2H4O2")           #Schnellauswahl für Essigsäure (rein)
elif Stoffauswahl == "5":
    Stoff = formula("C2H5OH")         #Schnellauswahl für Ethanol
else:
    Stoff = formula(Stoffauswahl)     #Beliebige Formel

#Serielle Kommunikation:
ser = serial.Serial(port='/dev/ttyS0', baudrate=9600, bytesize=serial.EIGHTBITS, parity=serial.PARITY_NONE, stopbits=serial.STOPBITS_ONE, timeout=.5, xonxoff=False, rtscts=False, write_timeout=None, dsrdtr=False, inter_byte_timeout=None, exclusive=None) #Kommunikation wird eröffnet
ser.write(b'S \r\n')                 #Aufforderung an Waage zum Ausgeben des Gewichts
time.sleep(0.5)                       #Kurze Wartezeit
i = 0
while i < 1 :
    Antwort = (ser.read_until(expected=b'g')) #Auslesen und Festhalten des Rückgabewertes der Waage
    if len(Antwort) > 14:               #Falls die nötige Zeichenlänge erreicht ist...
        i=1                             #...wird der Vorgang beendet
Antwort = Antwort.decode("utf-8")      #Decodierung der Antwort

#Start und Endpunkte werden festgelegt, je nach Anzahl Stellen:
Ende= ((Antwort.find("g"))-1)          #Festlegen des gleichbleibenden Endes
Liste= ["1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-"] #Liste zum Prüfen der Endpunkte wird erstellt
if Antwort[7] in Liste :               #Falls von links nach rechts an Stelle 7 das erste Mal ein Zeichen steht, ...
    Anfang= 7                           #...so ist Zeichen Nr. 7 der Anfangspunkt
elif Antwort[8] in Liste :             #Falls nicht, ist es Stelle 8?
    Anfang= 8
elif Antwort[9] in Liste :             #Und so weiter
    Anfang= 9
elif Antwort[10] in Liste :
    Anfang= 10

#Verschiedene Umwandlungen und Ausgaben:
Masse= float(Antwort[Anfang:Ende])     #Extrahieren der Zahl aus der Waagen-Antwort
print(str(Masse), " g")                 #Anzeigen der Masse in Gramm
molare_Masse = Stoff.mass                #Aufrufen der molaren Masse
Stoffmenge = float(Masse)/molare_Masse #Errechnen Stoffmenge
print(round(Stoffmenge, 3), " Mol")     #Ausgabe der gerundeten Stoffmenge

```

6.6 Dokumentinfo

Titel der Arbeit	Molwaage
Untertitel	Entstehung eines Prototyps
Fach	Chemie / Informatik
Dokumentname	MA2021_MolwaageEntstehungeinesPrototyps_FelixBrugnoni
Autor	Felix Brugnoni
Betreuungsperson	Dr. Rainer Steiger
Adresse 1	Kantonsschule Schaffhausen, Pestalozzistrasse 20, 8200 Schaffhausen
Abgabedatum PDF	29. Dezember 2021
Abgabedatum Bibliothek	4. Januar 2022