

Implementierung eines Mess- und Datenbanksystems fuer ein Fluoreszenzphotometer zur Bestimmung einer Malondialdehyd Stoffmengenkonzentration.

Implementation of a measurement- and database system for a fluorescence photometer to determinate the Malondialdehyd molarity of a specimen.

Christian Krause

Masterarbeit

Beginn der Arbeit:	14. Februar 2018
Abgabe der Arbeit:	14. August 2018
Erstgutachter:	Prof. Dr. Dieter Schumacher
Zweitgutachter:	Prof. Dr. Thomas Hohlfeld
Matrikelnummer:	1956616
Kontakt:	christian.krause@hhu.de

Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden.

Düsseldorf, den 14. August 2018

Christian Krause

Vorwort

Das passende Thema für eine Masterarbeit zu finden gestaltet sich mitunter schwierig. Es bestand der Wunsch zur Erarbeitung eines möglichst praxisorientierten Projektes, in welches insbesondere auch die jahrelang erarbeiteten IT-Kenntnisse der vorherigen Ausbildung einfließen sollten.

Bei der Absolvierung eines Abschlusses der medizinischen Physik darf aber natürlich auch die Medizin und Physik nicht fehlen. Einige Zeit war ich der Meinung, dass diese Anforderungen kaum vereinbar sind. Nach Rücksprache mit einigen Kommilitonen kristallisierte sich heraus, dass Herr Schumacher hier der richtige Ansprechpartner sei, und ich denke mit dem schlussendlich erarbeiteten Thema ist das breite Feld an Wünschen sehr gut abgedeckt.

Schon seit über 10 Jahren hat sich Linux bei mir als Desktopsystem etabliert, mit dem ich täglich arbeite. Schon vor dem Studium habe ich angefangen, erfolgreich einige Projekte auf Basis von Linux und Embedded Devices durchzuführen. Angefangen bei der Funktionserweiterung für NAS-Devices um spezifische Dienste über Werbeanruf-Filter für die Fritzbox bis hin zum Aufsetzen von VPN-Tunnelsystemen auf Basis des Raspberry Pi sind sowohl kommerzielle als auch private Projekte auf Linux/Embedded Devices durchgeführt worden. Das Angebot von Herrn Schumacher, ein Projekt auf Basis des Raspberry Pi als Spezialisierung für die Masterarbeit fertig zu stellen, welches darüber hinaus den besagten medizinischen und physikalischen Hintergrund aufweist, war insofern äußerst interessant. Gleiches gilt für das daran anschließende Thema dieser Masterarbeit.

Ein herzliches Dankeschön an alle, die mich bei dieser Masterarbeit unterstützt haben. Neben Prof. Dr. Schumacher als Betreuer ist hier auch Yannick Joppien zu nennen, der ein sehr ausgereiftes Vorgängerprojekt hinterlassen und eine gute Projektübergabe gegeben hat. Ebenfalls ein Dank an Prof. Dr. med. Hohlfeld und Frau Bartkowski, die mich kompetent in allen medizinischen Aspekten des Projektes beraten haben, sowie an Herrn Rohrbach aus der Fachwerkstatt für die Arbeiten am Gehäuse.

Ein weiterer, nicht zu verachtender Dank geht an Herr Peters für die Erläuterungen zum Client Side Rendering und die Demonstration zur Nutzung von Websockets mit Node/Vue, um eine professionelle Oberfläche bieten zu können.

Darüber hinaus darf meine Frau Tanja an dieser Stelle nicht fehlen, die mir viele Monate Freiraum geschaffen hat, damit ich mich meiner Abschlussarbeit widmen kann, sowie Herrn Schütz, der das gleiche auf beruflicher Ebene ermöglicht (und erduldet) hat.

Inhaltsverzeichnis

1	Einleitung und Überblick	4
1.1	Motivation und Aufgabenstellung	4
1.2	Zeitliche Projektentwicklung	4
2	Vorüberlegungen	6
2.1	Auswahl eines geeigneten Computersystems	6
2.2	AD-Wandler: Auswahl und technische Funktion	7
2.3	Referenzspannungsquelle	8
2.4	Echtzeituhr (Real Time Clock)	8
2.5	Auswahl von Programmiersprache, graphischer Oberfläche und Datenbank	8
3	Realisierung der Hardware	10
3.1	Schaltungsdesign AD-Wandler und Referenzspannungsquelle	10
3.2	Grundinstallation von Raspbian	14
3.3	Grundkonfiguration des Raspbian-Systems, Installation und Autostart der Backend- und Frontend-Dienste	15
3.3.1	Backend-Installation	16
3.3.2	Frontend-Installation	17
3.4	Implementierung der Echtzeituhr (Real Time Clock)	18
3.5	Automatischer Start des Browsers	20
4	Verbesserungen am Gehäuse	22
5	Bedienungsanleitung	24
5.1	Anschließen und Einschalten des Gerätes	24
5.2	Einlegen / Wechseln der Probe	25
5.3	Graphische Bedienoberfläche	26
5.3.1	Messreihenansicht	26
5.3.2	Anlegen einer neuen Messreihe: Leerprobenerfassung	27
5.3.3	Anlegen einer neuen Messreihe: Kalibrierung	28
5.3.4	Messübersicht, Exportieren, Löschen	30
5.3.5	Messung durchführen	31
5.3.6	Nachträgliche Leerprobenerfassung	34
5.4	Ausschalten des Gerätes	37

6 Realisierung der Software	38
6.1 Backend	38
6.2 Datenbankkonzept	39
6.2.1 Haupttabelle main	39
6.2.2 Messtabelle measure	40
6.3 Frontend	42
6.3.1 layouts/default.vue	42
6.3.2 pages/index.vue	42
6.3.3 pages/calibrate.vue	43
7 Test-Messung	45
8 Fazit	47
8.1 Zusammenfassung	47
8.2 Ausblick	47
A Quellcode	48
B Abkürzungsverzeichnis	49
Literatur	51

1 Einleitung und Überblick

1.1 Motivation und Aufgabenstellung

Ziel der Arbeit ist die Implementierung einer computergestützten Bedienoberfläche an ein Fluoreszenzphotometer zur optischen Bestimmung der Malondialdehyd-Konzentration (MDA) einer Blutprobe. Das Gerät ermöglicht, die Gerinnungsfunktion des Blutes von Patienten mit einer kardiovaskulären Vorgeschichte (Schlaganfall, Myokardinfarkt) kostengünstig zu überwachen.

Kommt es im Blutkreislauf zu einer Gefäßläsion, so führt die Aktivierung der Thrombozyten im Blut zur Thrombozytenaggregation und somit zur Blutstillung am betroffenen Gefäß. Bei der Thrombozytenaktivierung entstehen Thromboxan A_2 (TXA_2) und MDA im Verhältnis 1:1. Die Korrelation der MDA-Konzentration mit der TXA_2 -Konzentration macht erstere zu einem indirekten Indikator der Blutgerinnungsfunktion[1], die sich kostengünstig mit opto-elektronischen Verfahren messen lässt. Zur Absenkung der Gerinnungsfunktion des Blutes wird Patienten, z.B. nach einer Operation, Acetylsalicylsäure (ASS) verabreicht, welche jedoch individuellen Wirkungsschwankungen unterliegt. Diese Schwankungen werden u.a. durch Wechselwirkung mit gängigen Schmerzmitteln (COX-1 Inhibitoren) hervorgerufen, darunter Ibuprofen, Naproxen und Metamizol (Novaminsulfon, Dipyron)[2]. Das macht eine regelmäßige Kontrolle der individuellen Wirkung von ASS in Einzelfällen erforderlich. Die Bestimmung von MDA im Blutplasma ist daher ein Weg zur Feststellung der individuell benötigten Dosierung von ASS oder Substituierung von ASS mit einem anderen Gerinnungshemmer zur Vermeidung von nachoperativen Thrombosekomplikationen.

Nachdem verschiedene messtechnische Aufbauten zum opto-elektronischen Nachweis von MDA in Blutproben von Daniel Schölzel verglichen und die präferierte Methode von Yannick Joppien implementiert wurde, wird dem dabei konstruierten Fluoreszenzphotometer mit dieser Arbeit eine computer- und datenbankgestützte Bedienoberfläche hinzugefügt, sowie abschließende Arbeiten am mechanischen Aufbau und dessen Unterbringung in einem praxistauglichen Gehäuse durchgeführt.[3][4]

1.2 Zeitliche Projektentwicklung

Der Messtechnik zugrunde liegt die Bildung eines Farbstoffmoleküls zum fluorimetrischen Nachweis: Im Blutplasma vorhandenem MDA wird Thiobarbitursäure (TBA) hinzugefügt, welches zusammen mit MDA nach halbstündigem Erhitzen auf 70°C das stabile Farbstoffmolekül Malondialdehyd-bis-(dimethylacetal) mit Absorptionsmaximum bei 532 nm und Emissionsmaximum bei 550 nm bildet[3]. Die Farbstoffkonzentration lässt sich in Abhängigkeit des im Plasma vorhandenen MDA nach entsprechender Anregung mit einem Laser oder einer Leuchtdiode photo- oder fluorimetrisch bestimmen.¹

¹Bei der Photometrie wird die Restlicht-Intensität nach Absorption durch das Farbstoffmolekül auf dem Lichtweg bestimmt. Bei der Fluorimetrie wird die Lichtintensität des Fluoreszenzleuchtens bestimmt, das vom Farbstoffmolekül ausgesandt wird. Die technische Herausforderung der Photometrie ist die Reproduzierbarkeit des Absorptionsverhalten; bei der Fluorimetrie ist es die Auskopplung des Anregungslasers durch Lochspiegel und passendem Sperrfilter.

Der medizinische Hintergrund sowie die Grundlagen zur Fluorimetrie werden in der Masterarbeit von Daniel Schölzel ausführlich beschrieben. Bezüglich methodischer Details wird daher auf diese Arbeit verwiesen.[4]

Die Arbeit von Daniel Schölzel zur opto-elektronischen Bestimmung der MDA-Konzentration im Blutplasma vergleicht drei verschiedene optische Aufbauten auf ihre messtechnische Eignung:

- Ein Differenzphotometer: Die Konzentrationsbestimmung erfolgt über die Messung der Absorption von grünem gegenüber rotem Licht aus einer Leuchtdiode, die sich unter gleicher Lichtaussendung nacheinander im gleichen Strahlengang befinden. Die Differenz der Absorption entspricht der Absorption durch den Farbstoff.
- Ein Fluorimeter mit 90° Detektoranordnung: Ein Laserstrahl beleuchtet die Probe. Ein Detektor mit Sperrfilter für die Wellenlänge des Laserlichts sowie ein Photomultiplier befinden sich in einem Winkel von 90° zum Laserstrahl. Der Detektor misst das Licht, welches aufgrund der fluoreszierenden Eigenschaft vom Farbstoff ausgesandt wird.
- Ein Fluorimeter mit Lochspiegel: Der Detektor mit Sperrfilter und der Photomultiplier befinden sich im Strahlengang des Laserstrahl. Der Laserstrahl selbst wird jedoch von einem Lochspiegel aus dem Strahlengang ausgekoppelt.[4]

Die Fluorimeter stellen sich aufgrund des verwendeten Photomultipliers gegenüber der beim Differenzphotometer verwendeten Photodiode als geeigneter in Bezug auf die Messgenauigkeit heraus, insbesondere im Hinblick auf die Genauigkeit bei niedriger Lichtintensität[4]. Die Lochspiegelanordnung wurde aufgrund der besseren Kompatibilität mit den im Labor verwendeten, nach unten verjüngten Küvetten gewählt, welche die Intensitätsbestimmung des 90°-Streulichtes erschweren[3]. Alle Messverfahren besitzen die Eigenschaft, dass innerhalb des Messbereiches ein linearer Zusammenhang der Messspannung zur Konzentration des MDA besteht[4], was die Kalibrierung vereinfacht.

Im Rahmen seiner Bachelorarbeit entwickelte Yannick Joppien eine Schaltung, welche die benötigten Betriebsspannungen für den Laser, den Photomultiplier und die Operationsverstärker, sowie die Referenz- und Controlspannung² des Photomultipliers zur Verfügung stellt. Darüber hinaus wurde ein Gehäuse entwickelt, welches die Messkomponenten gegen Tageslicht (Streulicht) abschirmt. In dieser Arbeit wurde die Funktionalität des Testgerätes auf Basis neuer Komponenten (Raspberry Pi und AD-Wandler) erweitert, sowie die Funktionalität des Gehäuses erhöht. Hierzu zählt eine Klappe für den einfachen Wechsel des Probenröhrchens.

²Die Controlspannung hat Einfluss auf die Beschleunigungsspannung im Photomultiplier und beeinflusst den Verstärkungsfaktor.

2 Vorüberlegungen

Die Aufgabenstellung wurde in Teilbereiche zerlegt und für die jeweiligen Bereiche ein Weg zur Lösung erarbeitet. Im Einzelnen ergaben sich folgende Kernaspekte:

1. Erfassung der Messwerte durch Ansteuerung eines AD-Wandlers³
2. Bereitstellen einer graphischen Oberfläche mit Messlogik
3. Erstellen einer Messwert-Datenbank mit Exportfunktion
4. Überarbeitung des Gehäuses gemäß einer Designvorgabe durch die Fachwerkstatt

2.1 Auswahl eines geeigneten Computersystems

Für die Realisierung musste ein Computersystem gewählt werden, welches sowohl über genügend Rechenleistung und Speicher verfügt, um die Datenbank und graphische Oberfläche bereitzustellen, als auch die Möglichkeit besitzt, einen AD-Wandler bzw. ein Multimeter zu konnektieren. Aus der Spezialisierung zu dieser Arbeit, der „Optimierung eines computergestützten Temperaturreglers zur thermischen Aufbereitung einer Probe“ [5], war dem Autor bereits die Ansteuerung der GPIO Ein- und Ausgänge des Raspberry Pi Mini-Computer bekannt, sodass die naheliegende Idee darin lag, den Raspberry Pi zu verwenden und über die GPIO Ports einen AD-Wandler zu betreiben.

Eine mögliche Alternative stellt der Mikrocontroller Arduino dar. Im Gegensatz zum Raspberry Pi besitzt der Arduino einen integrierten AD-Wandler mit kalibrierter Referenzspannung, bietet jedoch weder genug Speicher noch Rechenleistung zur Bereitstellung der graphischen Oberfläche oder des Datenbanksystems, sodass ein zusätzliches System (z.B. ein weiterer Computer) für die Implementierung benötigt wird.

Die weitere Alternative besteht in dem Einsatz eines Personal-Computers oder aus Platzgründen eines Notebooks. Dem Notebook fehlt es allerdings an GPIOs um einen AD-Wandler ansteuern zu können. Eine Alternative zu den GPIOs stellt der Anschluss eines computergesteuerten Multimeters z.B. über USB dar. Gegen ein Notebook sprechen der deutlich höhere Preis, die weitestgehend fehlende Dokumentation ähnlicher Projekte, sowie die deutlich größeren Abmessungen des Systems.

Für den Raspberry Pi spricht die große Menge an vorhandener Projektdokumentationen, die auf Selbstbau-Seiten im Internet leicht zu finden und gut dokumentiert sind; meist mit entsprechender Open-Source Software oder gemeinfreier Beispiel-Implementierung in verschiedenen Programmiersprachen. Auch der Preis und die Abmessungen des Gerätes passen besser zu dem Projekt. Daher fiel die Auswahl schlussendlich auf den Raspberry Pi.

³Die GPIO-Ports des Raspberry Pi dienen als digitale Ein- oder Ausgänge. Als Eingänge geben sie an, ob eine Spannung am GPIO-Port anliegt oder nicht, wobei die Abgrenzung zwischen Spannung / keine Spannung durch einen durch das System festgelegten Schwellwert bestimmt ist. Ein AD-Wandler misst dagegen einen analogen Spannungswert, d.h. er bestimmt die exakte Höhe der am Photomultiplier anliegenden Spannung.

2.2 AD-Wandler: Auswahl und technische Funktion

Grundsätzlich wird für das Messen der Ausgangsspannung am Photomultiplier ein AD-Wandler (Analog Digital Converter, ADC) benötigt. Mögliche Formen sind ein klassischer integrierter Schaltkreis (Integrated Circuit, IC), oder alternativ die Nutzung eines Multimeters mit Computerschnittstelle. Nachteil eines Multimeters ist die Größe und der Preis, sowie die Tatsache, dass es sich um ein autarkes Gerät handelt, d.h. es muss – ohne sich genauer mit der Thematik zu befassen – separat ein- und ausgeschaltet werden und benötigt eine separate Energieversorgung. Insofern wurde zuerst untersucht, ob sich ADC-ICs direkt mit einem Raspberry Pi ansteuern lassen und ob die Funktionen dieser ICs für den Einsatzzweck ausreichend sind.

Eine grundlegende Eigenschaft eines AD-Wandlers ist seine Bittiefe (Auflösung). Typischerweise liegt diese in der Größenordnung zwischen 8 und 16 Bit. Ein 12 Bit AD-Wandler unterteilt den Messbereich in $2^{12} = 4096$ Messintervalle. Die zu messende Spannung wird mit der Referenzspannung verglichen. Der AD-Wandler gibt den Intervallbereich zurück, in welchem die zu messende Spannung liegt. Hierbei entspricht eine Spannung von 0 V dem 1. Intervallbereich und eine Spannung, die genauso groß oder größer als die Referenzspannung ist, dem Intervallbereich 4096.

Je höher die Auflösung, desto genauer kann die zu messende Spannung angegeben werden, da mit steigender Anzahl an Messintervallen die Schrittweite eines Intervallbereiches kleiner wird. Um die zur Verfügung stehende Auflösung des AD-Wandler-Systems optimal zu nutzen, liegt die verwendete Referenzspannung idealerweise knapp oberhalb der höchsten zu messenden Spannung, unterschreitet diese jedoch nicht, da Spannungswerte oberhalb der Referenzspannung nicht mehr von der Referenzspannung differenziert werden können.

Gängige AD-Wandler kosten zwischen 2 und 4 Euro und werden mithilfe des SPI-Bus angesteuert, den der Raspberry Pi an den GPIO Ports 9 bis 11 (BCM Notation)[6] beherrscht. Gut dokumentiert sind Projekte mit den ADC-Typen MCP300X mit 10 Bit Auflösung sowie der MCP320X mit 12 Bit Auflösung, z.B. in [7] und [8]. Das X steht jeweils für die Anzahl der analogen Kanäleingänge, von denen für dieses Projekt lediglich einer Verwendung findet. Verfügbare Modelle besitzen 2, 4 oder 8 Eingänge, wobei die MCP AD-Wandler mit nur 2 Eingängen jeweils keinen gesonderten Referenzspannungseingang besitzen und stattdessen die Versorgungsspannung als Referenzspannung nutzen. Da die Versorgungsspannung lastabhängig im niedrigen, einstelligen Prozentbereich schwankt, erzeugt die Nutzung der Versorgungsspannung als Referenzspannung einen Messfehler in der gleichen Größenordnung. Ein separater Eingang einer Referenzspannung ist daher obligatorisch.

Die Schrittweite beträgt beim 12 Bit MCP3204 bei einer Referenzspannung von 2,5 Volt 0,61 mV, beim 10 Bit MCP3004 sind es knapp 2,5 mV. Typische Kalibrierungskurven aus der Arbeit von Yannick Joppien ergeben Regressionsgraden mit einer Steigung von 1 mV /nM. Die Bestimmung der MDA Konzentration erfolgt ganzzahlig in nM. Ein 12 Bit AD-Wandler ist mit 0,61 mV Intervallbreite ausreichend, ein 10 Bit AD-Wandler mit 2,5 mV Intervallbreite nicht.

Aufgrund der einfachen Möglichkeit der Ansteuerung des AD-Wandlers mittels vorkonfigurierter Module am SPI-Bus für den Raspberry Pi wurde der Vergleich mit einem deut-

lich größeren und teureren Multimeter an dieser Stelle nicht weiter verfolgt, sondern der AD-Wandler Typ MCP3204 verwendet, der 12 Bit Auflösung bietet und über einen separaten Eingang für eine Referenzspannungsquelle verfügt. Weitere Informationen zum MCP3204 findet sich im Datenblatt unter [9].

2.3 Referenzspannungsquelle

Die Ausgangsspannung des Photomultiplier liegt zwischen 0 V und 15 V. Mit dem vorgegebenen Messaufbau von Yannick Joppien liegen die gemessenen Spannungen bei typischen Proben zwischen 0 V und 1,2 V, für Kalibrierungszwecke überschreiten sie knapp 2 V. Daher wurde bei der Wahl der Referenzspannungsquelle auf den LM4040 mit einer Referenzspannung von 2,5 V zurückgegriffen. Diese bietet bei vertretbaren Kosten eine definierte Fehlertoleranz, die mit einem Grad A⁴ Bauteil bei 0,1 % und einer Temperaturabhängigkeit von 100 ppm / °C liegt. Hilfreiche Diskussionen über Referenzspannungsquellen waren an der Stelle [10].

2.4 Echtzeituhr (Real Time Clock)

Der Raspberry Pi besitzt keine batteriebetriebene Echtzeituhr und verliert daher mit dem Ausschalten des Gerätes die aktuelle Uhrzeit. Das Raspbian Betriebssystem versucht diesen Umstand bestmöglich zu kompensieren. Standardmäßig wird nach jedem Hochfahren über das NTP Protokoll eine Zeitabfrage im Internet durchgeführt, sodass bei bestehender Internetverbindung keine RTC benötigt wird. Sofern keine Internetverbindung besteht, wird auf den Dienst fake-hwclock zurückgegriffen. Dieser speichert die aktuelle Uhrzeit stündlich in einer Datei und lädt diese Uhrzeit beim Neustart.[11] Das ist hilfreich, um bei Offline-Systemen einen stetig fortlaufenden Zeitfluss zu erhalten, sodass beispielsweise erstellte Datensicherungen eine sinnvolle, chronologische Reihenfolge erhalten; für dieses Projekt ist das allerdings nicht ausreichend, da die aktuelle Uhrzeit in der Datenbank zu Dokumentationszwecken abgespeichert wird.

Daher wurde eine externe Echtzeituhr Modell ds3231 über den I2C-Bus via GPIO angebunden. Durch die Nutzung des I2C-Bus ist keine unerwünschte Beeinflussung mit der Ansteuerung des AD-Wandlers via SPI-Bus zu erwarten. Die angegebene Echtzeituhr kostet wenige Euro, kann direkt auf die GPIO-Leiste des Raspberry Pi gesteckt werden und es finden sich viele Anleitungen, das Gerät im Raspberry Pi einzubinden. Insofern wurde nicht weiter mit anderen Modellen verglichen.

2.5 Auswahl von Programmiersprache, graphischer Oberfläche und Datenbank

Die Anforderungen an die Programmiersprache ergeben sich aus den folgenden Punkten:

- Die Ansteuerung des AD-Wandlers soll über einzubindende, vorgefertigte Module

⁴Fehlertoleranzklasse

realisiert werden, um die vergleichsweise komplexe, langwierige und fehlerträchtige Programmierung zur Ansteuerung externer Peripheriegeräte abzufangen und durch bewährte Bibliotheken zu ersetzen. Hierbei spielt auch die Güte der vorhandenen Dokumentation eine Rolle.

- Eine Interpretersprache ist von Vorteil, um Code schnell ohne langwierige Kompilierung auf dem prozessortechisch langsamen Raspberry Pi zu testen. Das ist insbesondere hilfreich, sofern nur kleine Änderungen am Quellcode vorgenommen und unmittelbar getestet werden.
- Die Anbindung einer Datenbank ist erforderlich, bevorzugt über vorimplementierte Schnittstellen.
- Die Möglichkeit zur Programmierung eines graphischen Interface muss vorhanden sein und die Einarbeitung in realistischer Zeit erfolgen können.

Da insbesondere die Verfügbarkeit externer Bibliotheken für die Anbindung einer Datenbank sowie des AD-Wandlers mitsamt Dokumentation für Python gewährleistet waren, wurde – analog zur Spezialisierung – Python als Programmiersprache gewählt, im Gegensatz zur Spezialisierung wurde jedoch die aktuelle Version Python 3 eingesetzt.

Die in der Datenbank gespeicherte Datenmenge ist sehr gering und auf wenige Tabellen beschränkt, die einzeln abgefragt werden, ohne dass Inhalte von Tabellen miteinander verknüpft werden („join“). Die Datenbank sollte sich mit SQL-Befehlen ansprechen lassen und in Anbetracht der knappen Systemressourcen des Raspberry Pi sehr sparsam in Bezug auf den Arbeitsspeicherbedarf sein. Alle diese Punkte werden von Sqlite erfüllt. Die Exportfunktion soll, da die Daten auf eine Tabelle und ein Bild beschränkt sind, über einen CSV-Export sowie den Export einer Graphikdatei erfolgen.

Die graphische Oberfläche lies sich mit Python nicht in gewünschter Form realisieren, daher wurde, nach Rücksprache mit Webentwicklern, das graphische Frontend von dem Backend mit Datenbankanbindung und Ansteuerung des AD-Wandlers getrennt und als Webanwendung mit dem Web-Framework *Vue*⁵ realisiert.

Ein Webframework ist eine Software, mit der sich dynamische Webanwendungen entwickeln lassen. Das Framework liefert viele häufig benötigte Funktionen bereits mit. Dadurch werden Code-Dopplungen vermieden und die Sicherheit gegenüber einer Eigenentwicklung erhöht. Es können sehr schnell lauffähige und komplexe Webanwendungen erstellt werden und durch den geringeren Anteil an eigenem Source-Codes wird die Selbst-Dokumentation erhöht.

Vue[12] steht in Konkurrenz zu großen Frameworks wie Angular (Google) und React (Facebook). Man sagt Vue nach[13], dass es eine sehr schnelle Lernkurve bei der Programmierung ermöglicht und im Vergleich zu Angular oder React verhältnismäßig schlank und schnell ist – eine ideale Voraussetzung für Programmierneinsteiger auf einem Embedded Device. Als Webserver kommt der Node-Webserver zum Einsatz, weil der Autor in die Kombination von Vue/Node-Webserver von einem Webentwickler eingewiesen wurde, sich diese Kombination als gut funktionierend herausgestellt hat und die im Web auffindbare Dokumentation zum Zwecke des Selbststudiums äußerst umfangreich ist.

⁵Aussprache wie das englische Wort „view“

3 Realisierung der Hardware

3.1 Schaltungsdesign AD-Wandler und Referenzspannungsquelle

Als Referenzspannungsquelle wird ein LM4040 im TO-92 Gehäuse verwendet. Der Gehäusetyp ist sowohl geeignet, eine Test-Schaltung auf ein Bread-Board zu stecken, als auch für spätere Hand-Lötarbeiten, für die sich alternativ verfügbare SMD-Bausteine nicht eignen.

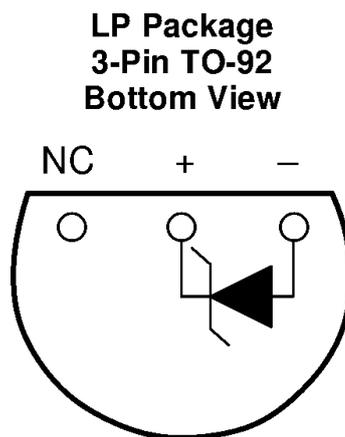


Abbildung 1: Pin Beschaltung des LM4040 laut Datenblatt [14]

Die Grundsaltung zur Bereitstellung der 2,5 V Referenzspannung stammt aus dem Datenblatt. Die Schaltung bedient sich der 3,3 V Eingangsspannung des Raspberry Pi und schaltet erst den Vorwiderstand, dann den LM4040 als Z-Diode in Serie gegen Masse. Zwischen Vorwiderstand und Z-Diode können dann – aufgrund des stabilen Spannungsabfall an der Z-Diode – exakt 2,5 V abgegriffen werden.

Shunt Reference Application Schematic

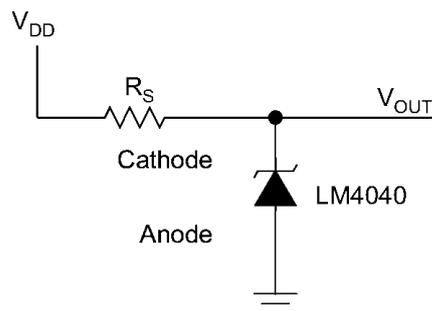


Abbildung 2: Basis-Schaltung für Referenzspannungsquelle aus dem Datenblatt [14]. Entsprechend der Quelle entspricht das Widerstands-Symbol dem ANSI-Standard.

Als AD-Wandler-Chip wurde der MCP3204 eingesetzt. Die Kommunikation mit dem Raspberry Pi erfolgt über den 4adrigen SPI-Bus. Der Raspberry Pi verfügt über 4 GPIO

Ports, die dieses Protokoll in Hardware beherrschen. Die verwendete Bibliothek GPIO-Zero kann den SPI-Bus darüber hinaus über andere GPIOs in Software emulieren, zu Lasten der System-Performance. Von dieser Funktion wurde kein Gebrauch gemacht. Die Dokumentation zum Anschluss und der Nutzung der Bibliothek findet sich auf der Internetseite der GPIO-Zero-Bibliothek [15].

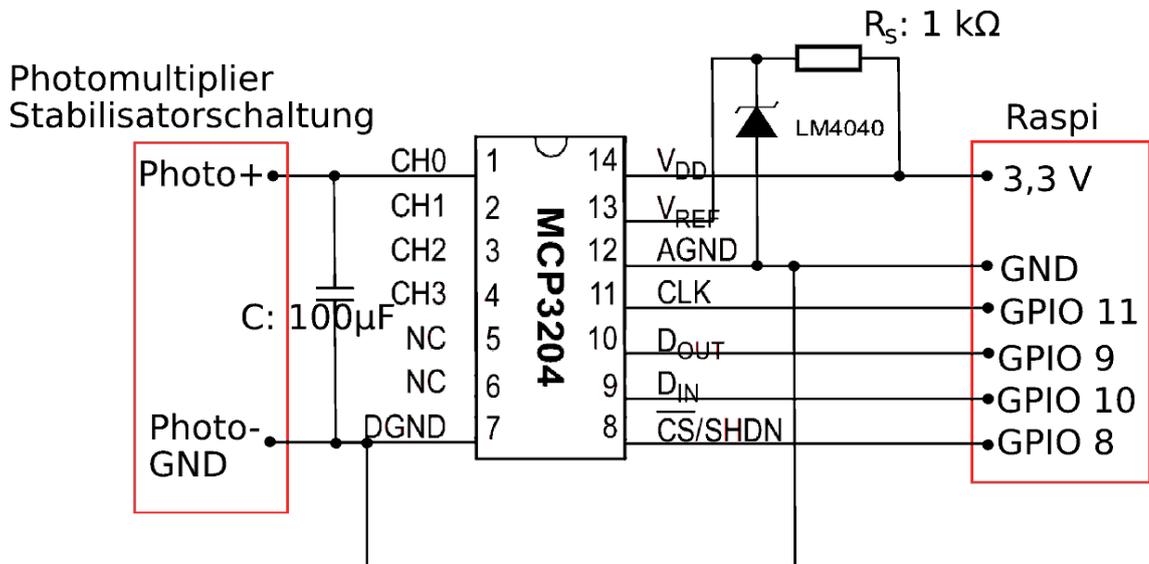


Abbildung 3: Schaltungslayout des Analog/Digital-Wandlers MCP3204, Angaben der GPIOs in BCM-Notation. Der Spannungsabgriff am LM4040 liefert 2,5 V Referenzspannung.

Die vier Adern des SPI-Bus beinhalten jeweils eine Leitung für die Datenübertragung in beide Richtungen (Data in, Data out), ein Taktsignal (Clock) zur Zeitsynchronisation der Datenübertragung, sowie eine Leitung zur Auswahl des analogen Kanaleingang (Channel Select), der zu messen ist.

Die beiden Masse-Anschlüsse für den analogen und digitalen Teil des AD-Wandlers werden zusammengelegt und mit der Masse des Raspberry Pi verbunden. Die Versorgungsspannung wird mit der 3,3 V Leitung des Raspberry Pi verbunden. Alternativ ist auch der Anschluss an einen 5 V Pin möglich.

Die Ausgänge der Photomultiplier-Stabilisatorschaltung werden mit dem verwendeten analogen Kanal 0 und der Masse verbunden. Mittels Kondensator zwischen dem Kanal 0 und der Masse in der Größe $100 \mu\text{F}$ wird das Ausgangssignal des Photomultipliers zusätzlich geglättet.

Die Schaltung wurde zuerst auf einem Breadboard getestet und im Anschluss gelötet.

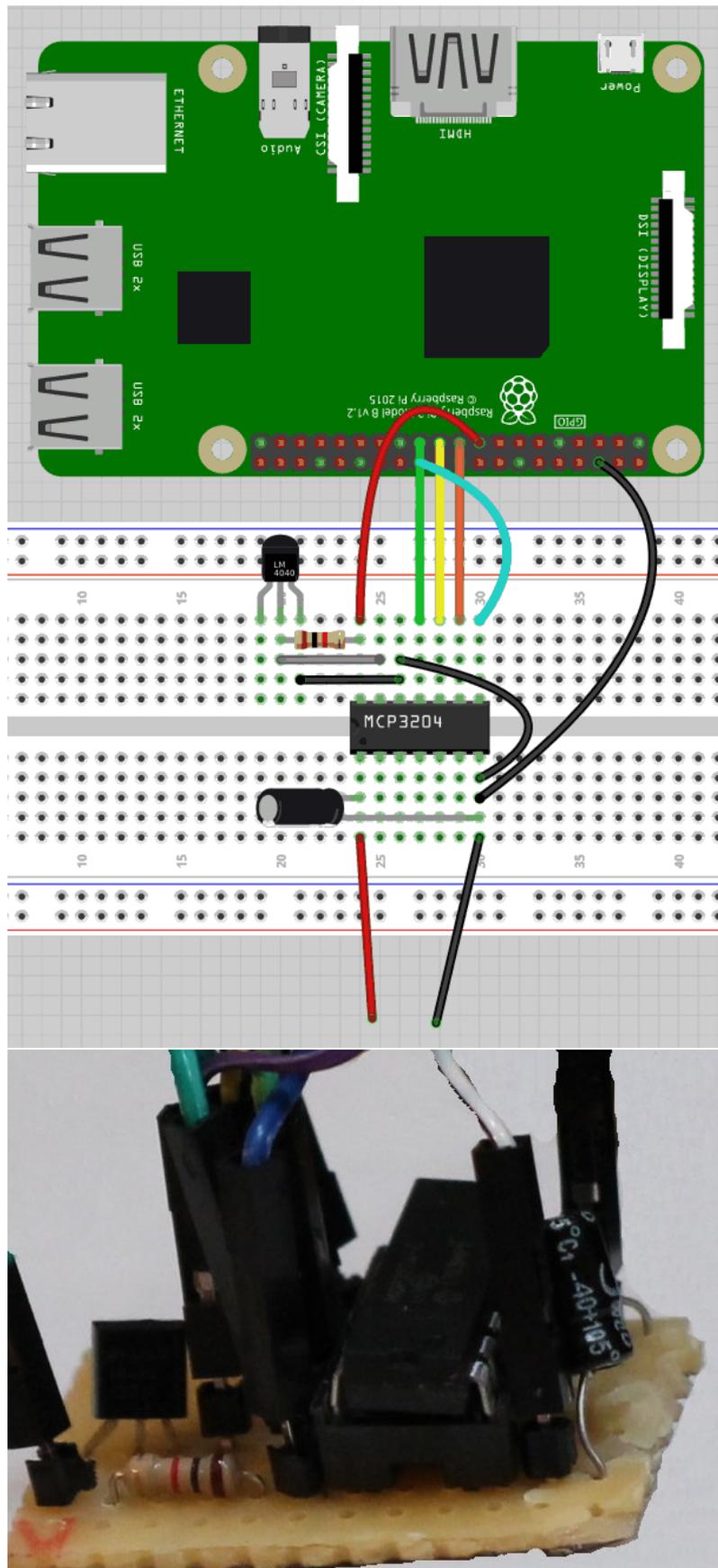


Abbildung 4: Breadboard Schaltungsdesign und gelötete Platine

Glättung des Ausgangssignals



Erste Messungen wurden mit einer Gleichstromquelle durchgeführt, mit welcher der AD-Wandler gute und genaue Werte liefert. Mit der Durchführung von Messungen am Ausgang der Stabilisatorschaltung des Photomultipliers traten deutliche Abweichungen zu Messungen mit einem Multimeter auf, insbesondere ergaben sich zeitliche Schwankungen bei der Messung. Die Bildschirmausdrucke eines Oszilloskop zeigen deutliche Wechselstromanteile.

- Oben links: Ohne Kondensator: Hoher Wechselstromanteil und Störspitzen
- Oben rechts: 100 nF Keramik-Kondensator
- Unten links: 2x 100 nF Keramik-Kondensator in Parallelschaltung
- Unten rechts: 100 μ F Elektrolyt-Kondensator

Schlussendlich wurde auf die AD-Wandlerschaltung ein zusätzlicher 100 μ F Elektrolyt-Kondensator zur Glättung des Wechselstromanteils und zur Filterung der Störspitzen gelötet.

3.2 Grundinstallation von Raspbian

Zuerst wird das Abbild des auf Debian basierenden Betriebssystems Raspbian heruntergeladen [16]. Hierbei wird die Version mit graphischer Oberfläche gewählt. Im Anschluss wird die Datei entpackt und mittels Linux-Kommando `dd` auf die Speicherkarte kopiert.

Kopieren des entpackten Raspbian-Abbild auf die Speicherkarte mittels `dd`

```
sudo dd if=~/.2018-06-27-raspbian-stretch.img of=/dev/sdb bs=1M
```

Hierbei gibt `/dev/sdb` den Pfad der Speicherkarte an, d.h. dieser Parameter muss angepasst werden, um nicht versehentlich einen falschen Datenträger unwiderruflich zu überschreiben. `bs` gibt die Blockgröße von 1 MB für den Schreibvorgang an. Wird dieser Parameter weggelassen, wird die anachronistische Standard-Blockgröße von 512 Byte verwendet. Da SD-Karten teilweise mit höheren internen Blockgrößen arbeiten, erhöht sich der Schreibaufwand ohne `Blocksize`-Angabe auf Faktor 2 bis 8. Mit noch höheren Blockgrößen kann der Schreibvorgang durch einen vergrößerten Schreibpuffer nochmals leicht beschleunigt werden. Die Blockgröße von 1 MB bewährt sich deshalb, weil die Umrechnung von Blöcken zur Datenmenge in MB dann besonders einfach fällt.

Nachdem der Kopiervorgang abgeschlossen ist, wird die Karte in den Raspberry Pi eingelegt. Mit angeschlossenen Eingabegeräten und Monitor bootet der Raspberry Pi. Da der Raspberry Pi zu Beginn des Bootvorganges den Monitor in Bezug auf die unterstützten Auflösungen konfiguriert, muss der Monitor vor dem Start eingesteckt sein. Raspbian passt die aus dem Abbild geklonte Betriebssystempartition beim ersten Start automatisch auf die Speichergröße der verwendeten SD-Karte an. In der neuesten Version von Juni 2018 werden nach dem ersten Start über einen Wizard weiterhin die Lokalisation für Spracheinstellungen, Zeitzone und Tastaturlayout abgefragt. Weiterhin wird an dieser Stelle angeboten, eine WLAN Verbindung herzustellen sowie Aktualisierungen zu installieren. Es bietet sich an, dem Wizard zu folgen.

Einstellungen im Wizard

Lokalisation: Germany / Timezone: Berlin

Die Standard-Zugangsdaten von Raspbian lauten:

Standard Zugangsdaten

Benutzer: pi
Kennwort: raspberry

Die Daten müssen zum Verbinden mit SSH oder VNC eingegeben werden. Sie werden nicht für den Desktop-Betrieb des Systems benötigt.

3.3 Grundkonfiguration des Raspbian-Systems, Installation und Autostart der Backend- und Frontend-Dienste

Nach der Grundinstallation können Aktualisierungen installiert werden. Darüber hinaus bietet es sich an, einige nicht benötigte Pakete zu entfernen. Auch Python 2 wird nicht benötigt und wird deinstalliert, da es – auch beim Autor dieser Thesis – regelmäßig zu Verwechslungen von Kommandozeilenbefehlen für Python 3 und deren Python 2 Pendanten kommt. Wichtig an dieser Stelle ist die Deinstallation der Python 3 Numpy Erweiterung, welche für die Berechnung der Regressionsgeraden verwendet wird. Die mit der verwendeten Raspbian-Version ausgelieferte Erweiterung erzeugte Versionsinkompatibilitäten mit anderen Erweiterungen, die via Python Installer pip⁶ heruntergeladen wurden. Daher wird Numpy über den Paketmanager apt-get / aptitude deinstalliert und später über pip3 nachinstalliert.

Aktualisieren der Paketquellen und löschen unbenötigter Pakete

```
sudo apt-get update
sudo apt-get remove wolfram-engine minecraft-pi \
sonic-pi scratch bluej python2* python3-numpy
sudo apt-get autoremove
sudo apt-get dist-upgrade
```

Nun werden Einstellungen über das Startmenü unter *Einstellungen* im Unterpunkt *Raspberry Pi Konfiguration* getätigt. Sofern die Sprache noch auf Englisch eingestellt ist, heißt der Eintrag *Preferences - Raspberry Pi Configuration*.

Einstellungen im Raspberry Pi Configuration Menü

- System – Underscan: Disabled (Entfernt schwarze Bildschirmränder)
- Interfaces: SSH aktivieren (optional)
- Interfaces: VNC aktivieren (Fernwartung, Nutzung ohne Eingabegeräte, opt.)
- Interfaces: SPI aktivieren (für AD-Wandler)
- Interfaces: I2C aktivieren (für Echtzeituhr)

Der Quellcode des Programmes wird auf den Raspberry Pi bevorzugt in das Benutzer-Verzeichnis kopiert. Der Kopiervorgang lässt sich z.B. mit einem USB-Stick oder über Netzwerk bewerkstelligen. Der Quellcode ist dabei unterteilt in den Backend-Teil, geschrieben in Python 3, der sich im Unterordner „ADpiServer“⁷ befindet, sowie den Frontend-Teil, entwickelt mit Vue/Node/Javascript unter „ADpiClient“.

Zuerst werden die Abhängigkeiten des Programmes aufgelöst, d.h. zum Starten notwen-

⁶nicht zu verwechseln mit pip für Python 2

⁷ADpi steht für Analog Digital raspberry PI

dige Bibliotheken und Laufzeitumgebungen werden heruntergeladen. Erfahrungsgemäß differieren die vorinstallierten Pakete mit jeder Raspbian-Version, daher wird hier Bezug auf Raspbian aus 06/2018 genommen.

3.3.1 Backend-Installation

Weitere Informationen: Pipenv-Umgebung für die Backend-Installation

Die ursprüngliche Idee zur Installation aller notwendigen Python 3-Bibliotheken für die Laufzeitumgebung des Backends lag in der Verwendung von Pipenv. Häufige Probleme bei der Nutzung von Python und anderen Programmierumgebungen liegen in der Inkompatibilität verschiedener Versionen gleicher Laufzeitbibliotheken. Im Worst-Case werden für verschiedene Programme auf einem System verschiedene Versionen von Bibliotheken benötigt, deren gleichzeitige Installation sich gegenseitig ausschließt. Pipenv stellt jedem Programm eine virtuelle Umgebung mit den passenden Bibliotheken zur Verfügung. Darüber hinaus wird die Installation der gewünschten Umgebung durch eine bereitgestellte Konfigurationsdatei auf dem Zielsystem automatisiert.

Leider kam es bei der Nutzung von Pipenv auf dem Raspberry Pi zu ungeklärten Problemen. Die Bibliothek GPIO-Zero erkannte innerhalb der Pipenv-Umgebung nicht mehr, dass sie auf einem Raspberry Pi ausgeführt wird und verweigerte dann die Ansteuerung der GPIOs. Andere Bibliotheken ließen den Raspberry Pi reproduzierbar abstürzen. Da nur ein einziges Python-Programm auf dem Gerät läuft, wurde daher schlussendlich auf die Pipenv-Umgebung verzichtet und die Bibliotheken manuell via pip3-Paketmanager installiert. Da bei der Vermischung von Bibliotheken aus dem pip3-Paketmanager und den vorinstallierten Paketen aus dem Raspbian-Software-Repository erneut Probleme auftraten, wurden etwaige Versionen aus dem Raspbian-Repository entfernt und schlussendlich alle Bibliotheken mit pip3 installiert.

Die Pipenv-Konfigurationsdatei liegt dem Programm dennoch bei.

Die Installation aller für das Backend benötigten Bibliotheken gelingt mit den folgenden Befehlen:

Installation der Python 3 Komponenten für das Backend und Programmstart

```
cd ~/ADpiServer
sudo pip3 install flask flask-socketio eventlet \
    matplotlib numpy gpiozero rpio
python3 app.py
```

3.3.2 Frontend-Installation

Vue wird in Kombination mit dem Webserver Node eingesetzt. Für das Nachladen von Erweiterungen für Vue⁸ wird mit Node der npm (Node Packet Manager) mitgeliefert, der das Nachladen anhand einer Konfigurationsdatei übernimmt. Dabei wird ein Gesamtpaket geschnürt, in welchem eigener Quellcode, Anweisungen und Konfigurationsdateien für den npm sowie heruntergeladene und installierte Erweiterungen des Node-Servers enthalten sind. Gemäß der Node-Dokumentation wird empfohlen, Node und npm jeweils mit User-Rechten (d.h. nicht mit Administratorrechten) zu installieren. Hier ergaben sich unter Raspbian Probleme mit der mitgelieferten, recht betagten Version des npm und Node. Sofern ein Update über die npm-eigene Update-Funktion durchgeführt wurde, wurden alte Raspbian-Repository-Bestandteile mit neuen Dateien aus dem Update vermischt. Das führte zu Problemen mit Schreibberechtigungen in verschiedenen Pfaden und schlussendlich zu einem Abbruch der Kompilierung. Letzten Endes lag die Lösung darin, den npm über die Raspbian-Repository-Paketquellen zu installieren, über die eigene Update-Funktion auf den neuesten Stand zu bringen und im Anschluss die Paketquellen-Version wieder zu löschen. Danach wurde der Pfad, in dem sich das heruntergeladene Update befand der Linux-Path-Variable hinzugefügt, sowie Zugriffsrechte für Ordner mit `chmod` gewährt, bei denen die Kompilierung aufgrund eines fehlgeschlagenen Zugriffs abbrach.

Die Bereitstellung des Frontend erfolgt mit:

Frontend kompilieren

```
# Node/npm-Installation via Raspbian-Repository und Update
sudo apt-get install npm
npm update npm -g
npm install -g n
n stable

# Entfernen der Repository Version des npm, hinzufügen
# der neuen Node/npm Version zur PATH-Variable
sudo apt-get remove npm
export PATH=/usr/local/n/versions/Node/10.6.0/bin:\$PATH

# Installieren aller benötigten Erweiterungen,
# bauen und starten des Node-Servers
cd ~/ADpiClient
npm install
npm run build
npm run start
```

Um beide Programme ohne Benutzerinteraktion mit dem Hochfahren des Raspberry Pi zu starten, müssen die folgenden Befehle in die `/etc/rc.local` vor der letzten Zeile (vor `exit 0`) eingetragen werden.

⁸Siehe Hauptkapitel „Realisierung der Software“

Frontend kompilieren

```
/usr/bin/python3 /home/pi/ADpiServer/app.py &  
cd /home/pi/ADpiClient  
/usr/local/bin/npm run start &
```

3.4 Implementierung der Echtzeituhr (Real Time Clock)

Für die Anbindung der RTC wurde nach der Anleitung aus [11] vorgegangen. Diese wird hier verkürzt dargestellt, sowie um die Funktion erweitert, mit jedem Systemstart die Uhrzeit der Echtzeituhr als Systemzeit festzulegen.

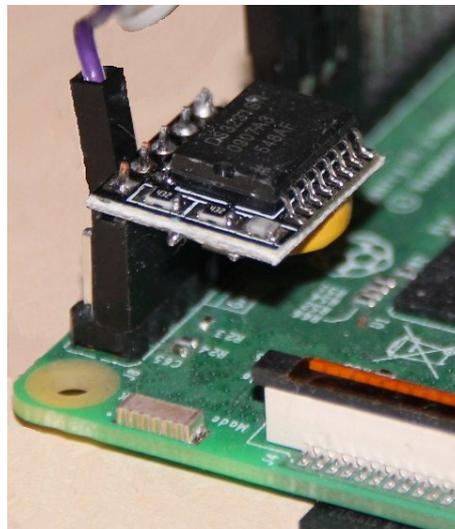


Abbildung 5: Das ds3231-Modul wird direkt auf die ungeraden GPIO-Ports 1 bis 9 gesteckt (Board-Notation)[6]

Im ersten Schritt werden die Module für den I2C-Bus aktiviert. Hierfür wird in den Raspberry-Pi Einstellungen der I2C-Bus aktiviert und die folgenden Module werden in die Datei `/etc/modules` eingetragen.

Module in die `/etc/modules` eintragen

```
i2c-bcm2708  
i2c_dev
```

Nach einem Neustart werden die I2C-tools installiert, sofern diese nicht bereits mit Raspbian ausgeliefert werden. Das war ab der Juni 2018 Ausgabe der Fall.

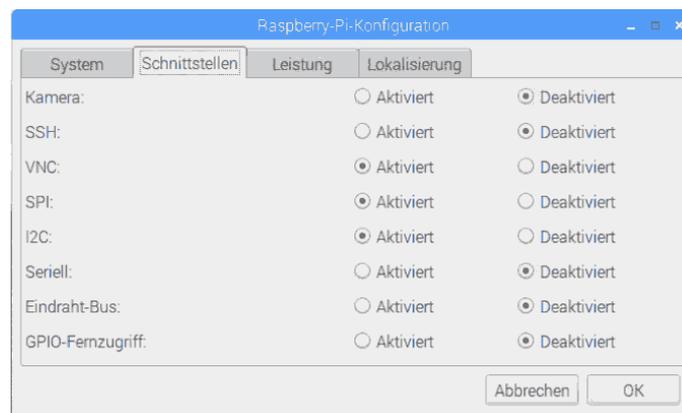


Abbildung 6: I2C-Bus aktivieren

I2C-Tools installieren

```
sudo apt-get install i2c-tools
```

Nun wird geprüft, ob sich die Echtzeituhr am I2C-Bus angemeldet hat.

Echtzeituhr überprüfen

```
# sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Die Echtzeituhr hat sich unter der Adresse 0x68h konfiguriert. Wir teilen dem Raspberry Pi mit, dass sich unter der angegebenen Adresse eine ds3231 Echtzeituhr befindet. Hierzu führen wir den unten stehenden Befehl einmal aus:

Dem Raspberry Pi die ds3231-RTC bekanntmachen

```
echo ds3231 0x68 | sudo tee /sys/class/i2c-adapter/i2c-1/new_device
```

Nun kann mit dem Befehl `sudo hwclock` auf die Uhr zugegriffen werden. Einmalig wird die Zeit der Hardwareuhr auf die aktuelle Systemzeit gesetzt:

Systemzeit auf die Echtzeituhr schreiben

```
sudo hwclock -w
```

Nach jedem Neustart soll die Echtzeituhr nun als Quelle der aktuellen Uhrzeit genutzt werden. Hierzu werden die folgenden Befehle in die Datei `/etc/rc.local` vor die Zeile `exit 0` am Dateiende geschrieben.

Systemzeit nach jedem Start aus der Echtzeituhr lesen

```
echo ds3231 0x68 | /usr/bin/tee /sys/class/i2c-adapter/i2c-1/new_device  
/bin/sleep 1  
/sbin/hwclock -s
```

Im letzten Schritt wird der NTP und der fake-hwclock Dienst deaktiviert.

Abschalten des NTP und der fake-hwclock

```
sudo update-rc.d ntp disable  
sudo update-rc.d fake-hwclock disable
```

Sofern zu einem späteren Zeitpunkt die Uhrzeit auf der Echtzeituhr korriert werden muss, kann dies bei aktiviertem Internetzugang mit dem folgenden Befehl geschehen.

Korrigieren der RTC-Uhrzeit via NTP

```
sudo ntpd -gq  
sudo hwclock -w
```

3.5 Automatischer Start des Browsers

Der Chromium-Browser soll beim Systemstart automatisch mitgestartet werden, hierfür wird der folgende Befehl in einem Terminal ausgeführt. Der Befehl hängt den Start-Befehl an das Ende der LXDE-Autostart Datei.

Automatischer Start des Chromium Browsers

```
echo @chromium-browser >> /home/pi/.config/lxsession/LXDE-pi/autostart
```

Zuletzt wird für den vorinstallierten Chromium-Browser die Startseite angepasst. Die URL des Frontend-Systems lautet:

Frontend-Webseite

```
http://localhost:3000
```

Zur Änderung der Startseite wird der Browser gestartet, die Einstellungen des Chromium Browser aufgerufen und die URL als Startseite eingetragen.

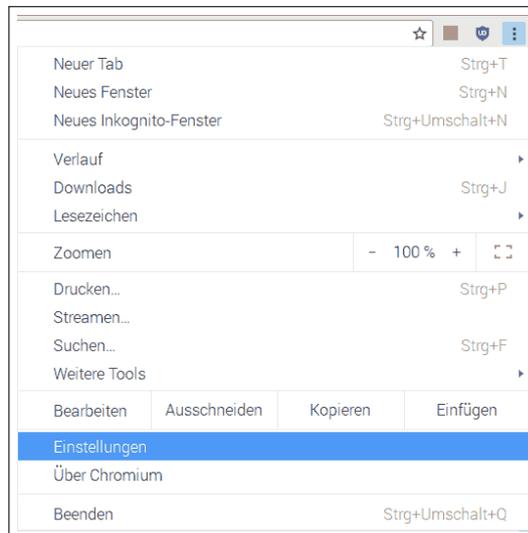


Abbildung 7: Einstellungen des Chromium-Browsers aufrufen...



Abbildung 8: ... und die URL hinzufügen, unter der das Frontend erreichbar ist

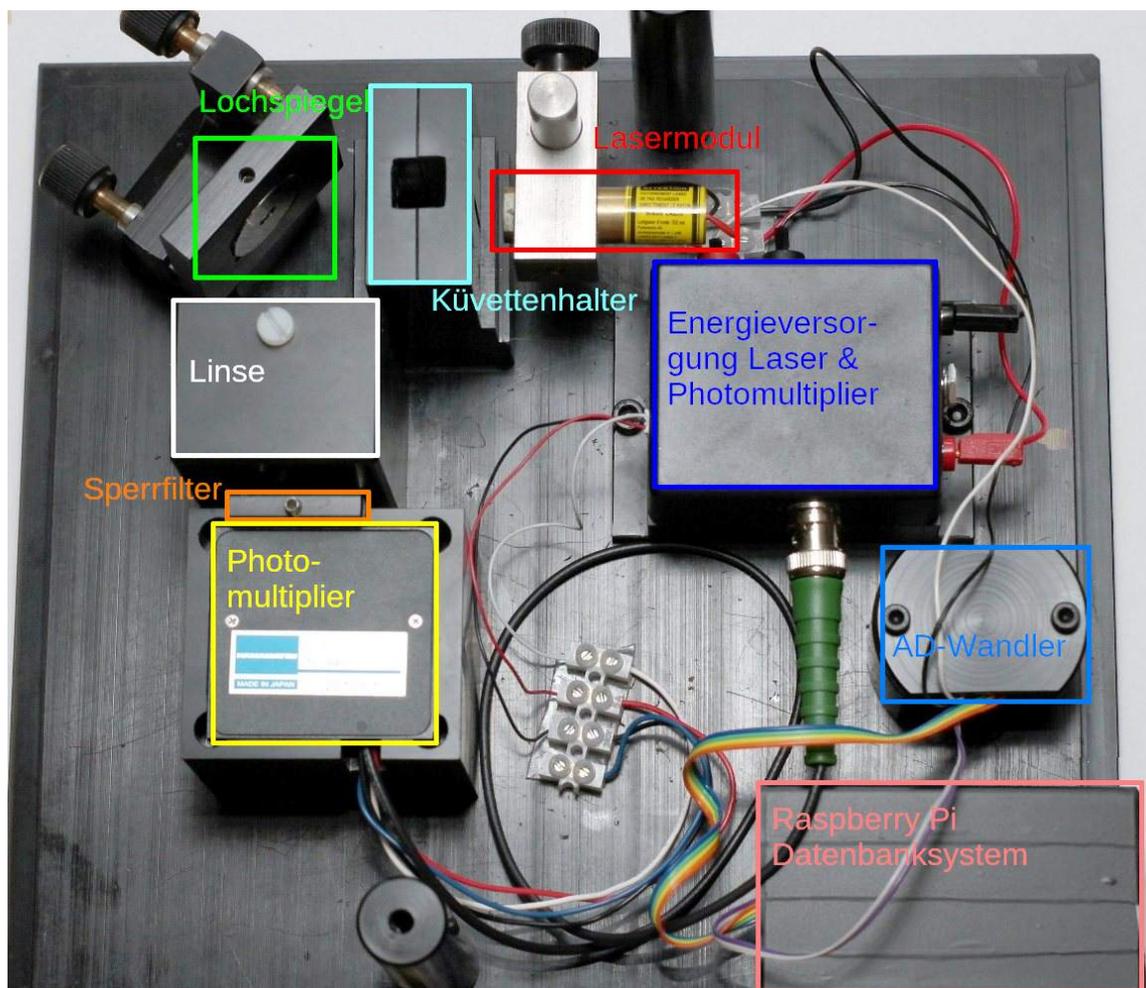
4 Verbesserungen am Gehäuse

Folgende Verbesserungen am Gehäuse wurden von Herrn Rohrbach aus der Fachwerkstatt gegenüber dem Stand von Yannick Joppien durchgeführt:

- In den Deckel der Messapparatur wurde eine Klappe integriert, die einen Zugang zur Probe gewährleistet, ohne das Gehäuse öffnen zu müssen.
- Die Höhe der Messapparatur wurde um ca. 4 cm verringert, sodass das Probenröhrchen bis unmittelbar unter die Klappe reicht. Der Wechsel der Probe durch die Probenklappe wird dadurch deutlich vereinfacht.
- Der Raspberry Pi wurde in einem eigenen Gehäuse so an den Rand der Grundplatte geschraubt, dass er nahtlos mit dem Gehäuse der Messapparatur abschließt. Hierdurch ist es möglich, den Raspberry Pi von außen mit Strom, Bildschirm und Eingabegeräten zu verbinden, ohne Kabel aus dem Gehäuse führen zu müssen. Lediglich das Stromkabel für den Laser/Photomultiplier muss nach wie vor über einen dafür vorgesehenen Schlitz aus dem Gehäuse geführt werden.



- Für den neuen AD-Wandler wurde ein kleines Gehäuse gefertigt, um diesen auf dem Gehäuse zu fixieren.
- Die Energieversorgung für Laser und Photomultiplier von Yannick Joppien wurde fest an die Grundplatte geschraubt.
- Die Position des Photomultiplier wurde verändert. Die vorherige Verschraubung verhinderte durch falsche Positionierung den Lichteinfall vom Lochspiegel durch die Linse.



5 Bedienungsanleitung

5.1 Anschließen und Einschalten des Gerätes

1. Schließen sie einen HDMI-fähigen Monitor mit einem HDMI-Kabel an den Raspberry Pi an. Alternativ nutzen sie ein HDMI-Kabel mit einem HDMI-DVI Konverter, um einen DVI-fähigen Monitor anzuschließen.
2. Stecken sie eine USB Maus- und Tastatur in die USB-Ports des Raspberry Pi.
3. Stecken sie das Micro-USB Netzteil in den Raspberry Pi. Schließen sie das Fluorimeter-Netzteil an das Fluorimeter an. Hierfür müssen sie das Gehäuse öffnen und das Kabel durch den Schlitz aus dem Gehäuse führen. Stecken sie beide Kabel in eine Steckdose.

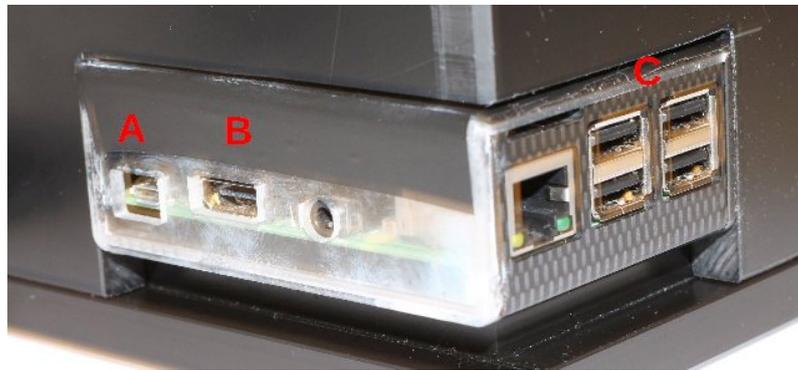


Abbildung 9: A:Micro-USB Stromanschluss für Raspberry Pi, B:HDMI-Bildschirmanschluss, C:USB-Anschluss für Tastatur/Maus und USB-Datenstick

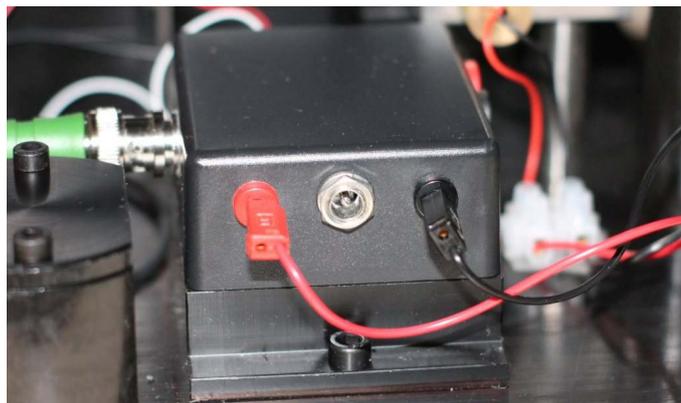


Abbildung 10: Stromanschluss Fluorimeter

Die Geräte starten automatisch. Einen eingeschalteten Raspberry Pi erkennt man an der links neben dem Micro-USB Anschluss aufleuchtenden, rot/grünen Diode im Inneren

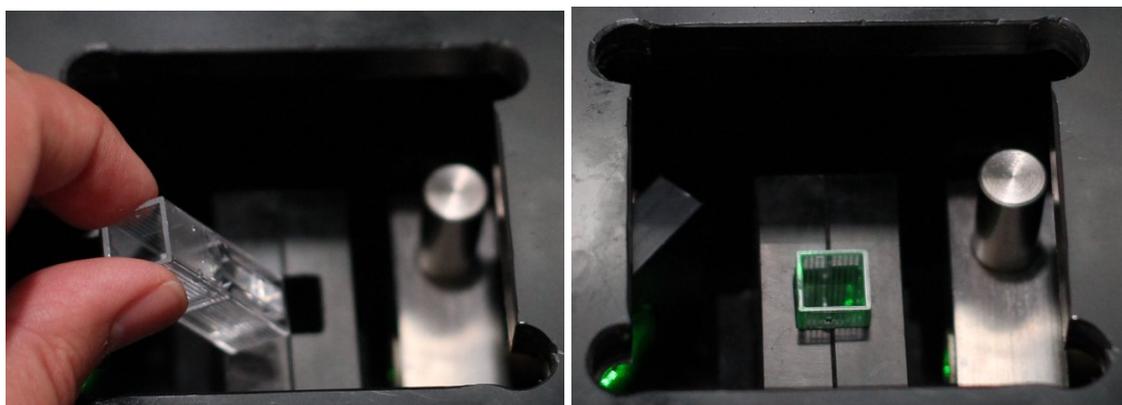
des transparenten Kunststoffgehäuses. Ein eingeschaltetes Fluorimeter ist an der grün beleuchteten Probe erkennbar. Nach kurzer Zeit ist die graphische Oberfläche des Raspberry Pi auf dem Monitor sichtbar und der Browser öffnet sich. Die Startseite des Browsers ist auf localhost : 3000 konfiguriert. Unter dieser URL ist das ADpi-Frontend erreichbar.

5.2 Einlegen / Wechseln der Probe

1. Öffnen sie den Probendeckel oben auf dem Gehäuse, um eine Probe in das Gerät einzulegen.



2. Achten sie beim Einlegen der Probe darauf, dass das Probenröhrchen so eingelegt wird, dass etwaige Verjüngungen nicht den Weg des Laserstrahl kreuzen. Probenröhrchen besitzen oft gegenüberliegende glatte Oberflächenseiten sowie gerillte. Der Laserstrahl, der von rechts nach links durch das Probenröhrchen läuft, muss die glatte Oberfläche passieren.



3. Verschließen sie den Probendeckel, bevor sie einen Messwert erfassen, damit kein Streulicht von außen während der Messung in das Gerät gelangt.

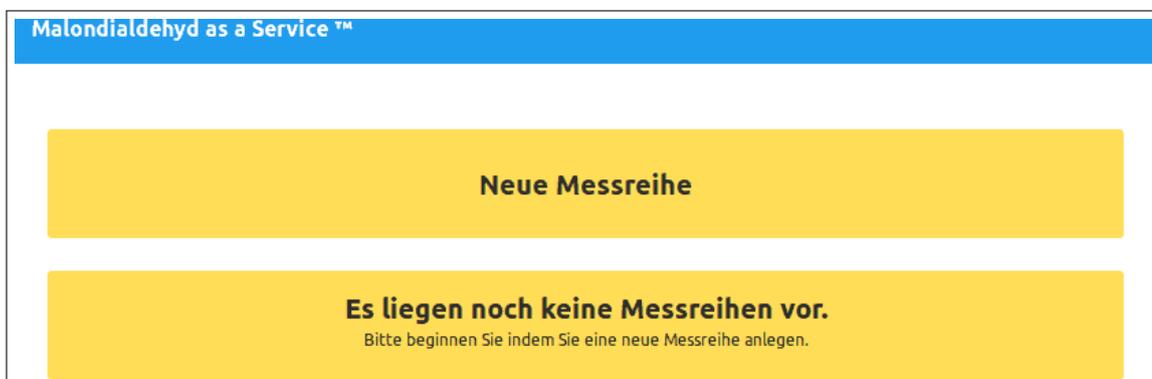
5.3 Graphische Bedienoberfläche

Die URL des Mess-Systems lautet `http://localhost:3000`. Die Seite ist als Startseite vorkonfiguriert und ist daher bereits geöffnet, wenn das Gerät gestartet wird.

5.3.1 Messreihenansicht

Zu Beginn liegen noch keine Messreihen vor. Mit einem Klick auf „Neue Messreihe“ wird eine neue Messreihe erstellt.

Ein Klick auf den oberen weißen Schriftzug „Malondialdehyd as a Service“ führt von überall auf die Messreihenansicht zurück. Achtung: Nicht gespeicherte Messergebnisse gehen hierbei verloren.



5.3.2 Anlegen einer neuen Messreihe: Leerprobenerfassung

Bei Erstellung einer Messreihe wird zu Beginn eine Kalibrierung durchgeführt, welche einer Messreihe fest zugeordnet ist. Die Kalibrierung beginnt mit einer Leerprobenerfassung. Die hier abgebildeten Messwerte sind keine echten Messwerte, sie wurden für den Frontend-Test simuliert.

The screenshot shows a web interface for creating a new measurement series. At the top, there is a text input field labeled 'Name der Messreihe' with a lock icon to its right. In the top right corner, the date and time '2018-08-13 17:57' are displayed, along with a share icon and a trash icon. Below the input field, the section 'Leerproben-Messtriple' is visible. It contains three measurement entries: '1. Messung: 0.21 V' with a red 'X' icon, '2. Messung: 0.22 V' with a red 'X' icon, and '3. Messwert erfassen' with a blue button. At the bottom of this section is a red button labeled 'Leerprobenmessung abschließen'. To the right of the measurement area is a large blue box titled 'Leerprobe erfassen' containing the text: 'An dieser Stelle wird ein Leerproben-Messtriple erfasst. Bitte geben Sie eine Leerprobe ohne Malondialdehyd in den Messplatz und klicken sie dann auf Messwert erfassen, um die Probe einzulesen.'

Die Leerprobenerfassung kann erst abgeschlossen werden, wenn alle drei Messwerte des Messtriple erfasst wurden. Dabei ist es dem Anwender überlassen, ob er eine Probe dreimal erfasst, oder die Probe für jede Erfassung getauscht wird.

The screenshot shows the same web interface as the previous one, but now the section is titled 'Testreihe'. The three measurement entries are: '1. Messung: 0.21 V' with a red 'X' icon, '2. Messung: 0.22 V' with a red 'X' icon, and '3. Messung: 0.22 V' with a red 'X' icon. Below these entries, the text 'Gemittelte Spannung der Leerprobe: 0.2167 V' is displayed. The blue button '3. Messwert erfassen' is now a blue button labeled 'Leerprobenmessung abschließen'. The right-side blue box 'Leerprobe erfassen' remains the same.

5.3.3 Anlegen einer neuen Messreihe: Kalibrierung

Im nächsten Schritt werden die Kalibrierungswerte erfasst. Hierfür werden Proben verschiedener Konzentrationen an Malondialdehyd erfasst. Die Konzentration wird hierfür vom Anwender festgelegt und jeweils ein Messtriple erfasst. Auch hier bleibt es dem Anwender überlassen, eine Probe mehrfach einzulesen oder für jede Messung eines Messtriples die Probe zu tauschen.

Testreihe 2018-08-13 17:57

Durchschnitt: 0.3267 V 100

1. Messung: 0.32 V X

2. Messung: 0.32 V X

3. Messung: 0.34 V X

Referenzprobe erfassen 200

1. Messwert erfassen

2. Messwert erfassen

3. Messwert erfassen

Referenzprobenmessung abschließen Weitere Referenzprobe +

Kalibrierungswerte erfassen

Nun werden Referenzwerte zur Erstellung einer Kalibrierkurve erfasst. Sie können beliebig viele Messtriple verschiedener Konzentrationen hinzufügen. Nach Abschluss der Erfassung können die Werte nicht mehr verändert werden. Die Referenzkurve wird dann berechnet und abgespeichert.

Die Kalibrierung kann durch Klicken auf „Malondialdehyd as a Service“ jederzeit abgebrochen werden. Es wird dann zur Messreihenansicht gewechselt.

Über „weitere Referenzprobe“ wird eine weitere Referenzmessung hinzugefügt. Über das Papierkorb-Symbol kann eine Referenzmessung gelöscht werden. Zum Abschluss der Kalibrierung muss mindestens eine Referenzmessung eingelesen werden. Darüber hinaus ist es möglich, eine Einzelmessung durch einen Klick auf das rote X neben dem Messwert zu annullieren und im Anschluss zu wiederholen.

The screenshot displays a software interface for entering calibration data. At the top, there is a text input field labeled 'Testreihe' and a date/time stamp '2018-08-13 17:57'. The main area is divided into three sections, each representing a different test series:

- Series 1:** Durchschnitt: 0.3267 V, Konzentration: 100. It contains three individual measurements: 1. Messung: 0.32 V, 2. Messung: 0.32 V, and 3. Messung: 0.34 V. Each measurement has a red 'X' button to its right for deletion.
- Series 2:** Durchschnitt: 0.4333 V, Konzentration: 200. It contains three individual measurements: 1. Messung: 0.44 V, 2. Messung: 0.43 V, and 3. Messung: 0.43 V. Each measurement has a red 'X' button to its right for deletion.
- Series 3:** Durchschnitt: 0.73 V, Konzentration: 500. It contains three individual measurements: 1. Messung: 0.73 V, 2. Messung: 0.73 V, and 3. Messung: 0.73 V. Each measurement has a red 'X' button to its right for deletion.

At the bottom of the interface, there are two buttons: 'Referenzprobenmessung abschließen' and 'Weitere Referenzprobe +'. On the right side, there is a blue box titled 'Kalibrierungswerte erfassen' with the following text: 'Nun werden Referenzwerte zur Erstellung einer Kalibrierkurve erfasst. Sie können beliebig viele Messtriple verschiedener Konzentrationen hinzufügen. Nach Abschluss der Erfassung können die Werte nicht mehr verändert werden. Die Referenzkurve wird dann berechnet und abgespeichert.'

Die Messung kann erst abgeschlossen werden, wenn:

- ein Name für die Messreihe vergeben wurde
- alle Konzentrationsangaben Zahlen > 0 enthalten
- in jeder Referenzmessung je alle drei Messungen erfasst wurden

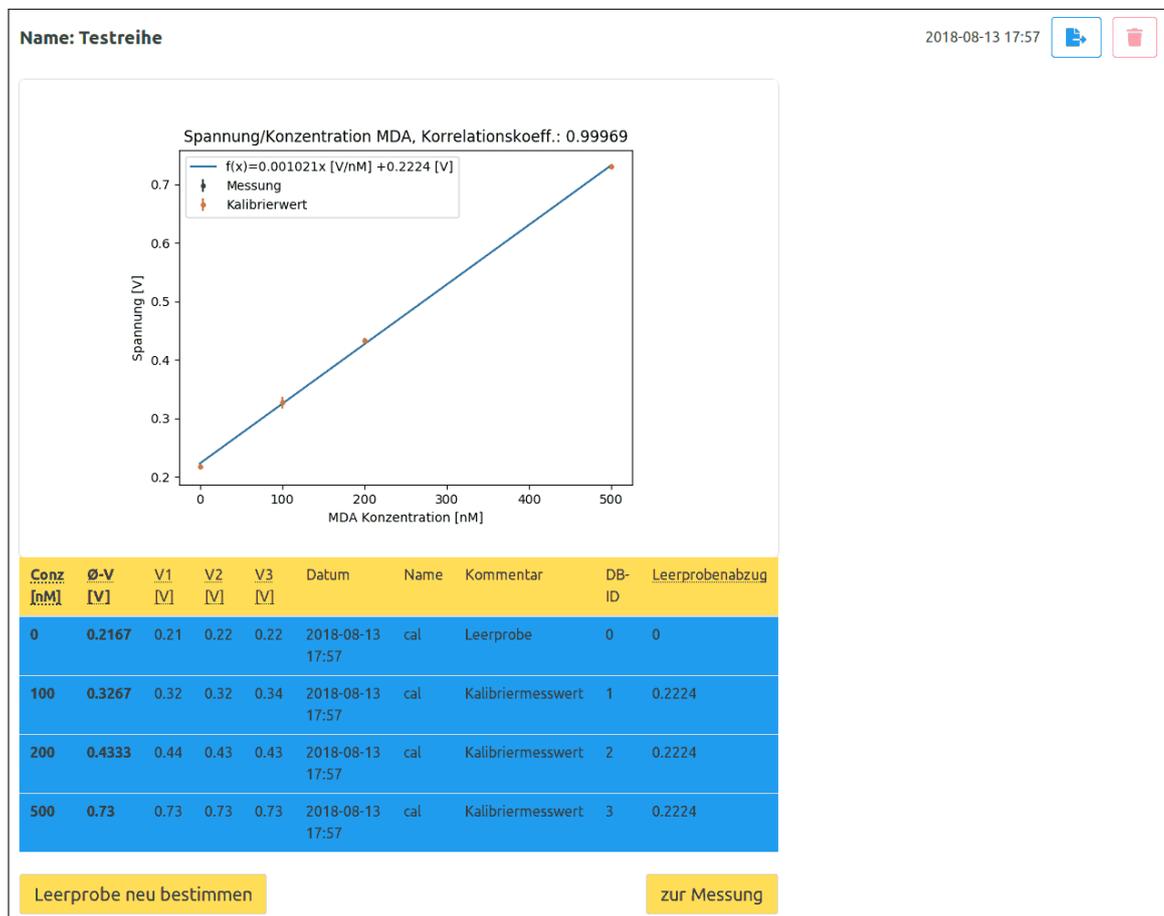
Der Messreihenname kann nach Abschluss der Referenzmessung nicht mehr geändert werden.

5.3.4 Messübersicht, Exportieren, Löschen

In die Messübersicht gelangt man nach Abschluss einer Kalibrierung, oder durch Öffnen einer bestehenden Messreihe.

Die Messübersicht enthält im oberen Bereich eine Abbildung, die die Kalibrierwerte als rote, sowie Messungen als grüne Punkte zeigt. Beide Werte enthalten Fehlerbalken. Darüber hinaus enthält die Abbildung einen Plot der Regressionsgeraden mitsamt der zugehörigen Geradenparameter und Korrelationskoeffizient. Bei der geplotteten Funktion handelt es sich immer um eine Gerade, d.h. um eine Funktion 1. Ordnung, da ein linearer Zusammenhang zwischen Malondialdehydkonzentration und Spannung angenommen wird.

Im unteren Bereich enthält die Messübersicht eine Tabelle aller Kalibrier- und Messwerte.



Über den Export-Button oben rechts wird die Messübersicht exportiert. Hierfür wird die Abbildung als PNG-Graphik, sowie die Tabelle als CSV-Datei auf einem angeschlossenen USB-Stick abgespeichert. Das Dateisystem des USB-Stick wird beim Einstecken automatisch eingebunden und nach erfolgtem Export automatisch ausgehängt, der USB-Stick kann dann herausgezogen werden.

Über den Lösch-Button am oberen Bildschirm wird die Messreihe gelöscht. Eine Löschung wird über eine Rückfrage bestätigt. Darüber hinaus kann die jeweils aktuelle Messreihe aus Sicherheitsgründen nicht gelöscht werden. Zum Löschen der aktuellen Messreihe muss erst eine neue, weitere Messreihe angelegt werden.

Das Export- und Lösch-Symbol der Messreihe ist nur in der Messübersicht verfügbar, nicht in der Messung, Leerprobenerfassung oder Kalibrierung.

5.3.5 Messung durchführen

Ein Klick in der Messübersicht auf den Button „zur Messung“ öffnet die Messwerterfassung.

Eine Messung erfasst die am Photomultiplier anliegende Spannung einer unbekanntenen Probe und berechnet über die bekannte Regressionsgerade der ausgewählten Kalibrierung die zugehörige Malondialdehyd-Konzentration.

Name: Testreihe 2018-08-13 19:10

Messwert erfassen Name

Kommentar

1. Messwert erfassen

2. Messwert erfassen

3. Messwert erfassen

Messungen speichern Weitere Messung +

Messwerte erfassen

Auf Basis der Kalibrierung dieser Messreihe können an dieser Stelle Messwerte erfasst werden. Hierfür legen sie eine Probe ein und messen diese drei Mal. Sie können entweder drei Proben des gleichen Patienten verwenden, oder alternativ eine identische Probe dreimalig erfassen.

Pflichtangabe für eine Messung ist ein Name. Darüber hinaus ist ein Kommentar optional. Die Messung muss auch hier als Messtriple erfolgen, d.h. es besteht die Möglichkeit, drei unabhängige, aber identische Proben zu messen, alternativ wird die selbe Probe dreimal hintereinander erfasst.

Name: Testreihe 2018-08-13 19:10  

 Konzentration: 247.6983 nM 



1. Messung: 0.478 V

2. Messung: 0.474 V

3. Messung: 0.474 V

Messwerte erfassen

Auf Basis der Kalibrierung dieser Messreihe können an dieser Stelle Messwerte erfasst werden. Hierfür legen sie eine Probe ein und messen diese drei Mal. Sie können entweder drei Proben des gleichen Patienten verwenden, oder alternativ eine identische Probe dreimalig erfassen.

Es können ebenfalls mehrere Proben nacheinander erfasst werden, ohne zwischenzeitlich auf die Übersichtsseite zu gelangen. Darüber hinaus können vor dem Speichern einzelne Messungen wieder gelöscht werden. Die Messung kann gespeichert werden, wenn für jede Messung ein Name festgelegt wurde und alle Messwerte vollständig erfasst sind.

Name: Testreihe 2018-08-13 19:10  

 Konzentration: 247.6983 nM 



1. Messung: 0.478 V

2. Messung: 0.474 V

3. Messung: 0.474 V

 Messwert erfassen 



Messwerte erfassen

Auf Basis der Kalibrierung dieser Messreihe können an dieser Stelle Messwerte erfasst werden. Hierfür legen sie eine Probe ein und messen diese drei Mal. Sie können entweder drei Proben des gleichen Patienten verwenden, oder alternativ eine identische Probe dreimalig erfassen.

Mit dem Diskettensymbol wird zwischen Eingabefeld und Anzeigefeld hin- und hergewechselt. Es dient als Schutz vor Fehleingaben, ein Klick hierauf ist jedoch Optional und für das Speichern der Messung nicht erforderlich.

The screenshot displays a software interface for entering measurement data. At the top left, it says "Name: Testreihe". On the top right, there is a timestamp "2018-08-13 19:56" and two icons: a blue diskette and a red trash can. The main area contains a "Messwert erfassen" button with a red trash icon, a "Testpatient" field with a blue edit icon, and a "Hier steht ein Kommentar" field with a blue edit icon. Below these are three buttons labeled "1. Messwert erfassen", "2. Messwert erfassen", and "3. Messwert erfassen". At the bottom left is a red "Messungen speichern" button, and at the bottom right is a blue "Weitere Messung +" button. A blue information box on the right contains the text: "Messwerte erfassen. Auf Basis der Kalibrierung dieser Messreihe können an dieser Stelle Messwerte erfasst werden. Hierfür legen sie eine Probe ein und messen diese drei Mal. Sie können entweder drei Proben des gleichen Patienten verwenden, oder alternativ eine identische Probe dreimalig erfassen."

Achtung: Datenschutz

Sofern bei einer Messung eine Blutprobe eines Patienten gemessen wird, handelt es sich bei den mit diesem Gerät erhobenen Daten um personenbezogene Daten gemäß Art. 4 der Datenschutzgrundverordnung (DSGVO), aufgrund der Gesundheitsmerkmale entsprechen die Daten im Speziellen den besonderen Kategorien von Daten gemäß Art. 9 DSGVO, die einem besonderen Schutz unterliegen.

Die DSGVO sieht im Art. 32 vor, dass Daten unter angemessener Sicherheit verarbeitet werden und räumt darüber hinaus dem Betroffenen weitgehende Rechte ein, darunter je nach Umständen der Verarbeitung die Löschung seiner Daten gem. Art 17 DSGVO.

Eine Löschung von einzelnen Messungen ist mit diesem Gerät nicht möglich, es können nur ganze Messreihen entfernt werden. Darüber hinaus ist die angemessene Sicherheit der Verarbeitung im Hinblick auf die Anforderungen der DSGVO u.U. nicht erfüllt, da das Gerät ohne Benutzerauthentifizierung arbeitet und ein Export der Daten für jeden Bediener möglich ist.

Daher wird an dieser Stelle empfohlen, keine Patientennamen im Namensfeld zu verwenden, sondern stattdessen mit Patienten-IDs zu arbeiten. Im Zweifel kann dann bei einer Löschanforderung durch das Entfernen der Patienten-ID aus der Patientenakte eine Anonymisierung erfolgen, da dann eine Zuweisung der IDs zu den Patientenstammdaten nicht mehr möglich ist. Anonymisierte Daten sind keine personenbezogene Daten im Hinblick auf Art. 4 bzw. Art. 9 der DSGVO, da ein Personenbezug nicht hergestellt werden kann.

5.3.6 Nachträgliche Leerprobenerfassung

Aus der Messübersicht heraus kann eine neue Leerprobe erfasst werden. Die Leerprobe wird als separate Messung angezeigt und besitzt nur temporäre Gültigkeit während einer Sitzung. Die Leerprobe wird bei erneutem Aufruf der Messreihe über die Messreihenansicht verworfen. Um die Konzentrationsberechnung transparent zu gestalten, wird für jede Messung der zugrundeliegende Leerprobenabzug angegeben. Sofern keine nachträgliche Erfassung einer Leerprobe erfolgt, wird als Leerprobenabzug der Achsenabschnitt der Kalibrierungsfunktion verwendet, der bei korrekter Kalibrierung dem ursprünglichen Leerprobenwert (nahezu) entsprechen sollte.

Weicht der erfasste Leerprobenabzug vom ursprünglichen Leerprobenabzug um mehr als 10% ab, wird eine Warnmeldung ausgegeben, in der eine Neukalibrierung empfohlen wird, also das Anlegen einer neuen Messreihe. Die Meldung hat nur warnenden Charakter und beeinflusst oder verhindert eine Messung nicht.

Name: Testreihe 2018-08-13 19:12  

Leerproben-Messtriple

1. Messwert erfassen

2. Messwert erfassen

3. Messwert erfassen

Leerprobenmessung abschließen

Leerprobe erfassen

An dieser Stelle wird ein Leerproben-Messtriple erfasst. Bitte geben Sie eine Leerprobe ohne Malondialdehyd in den Messplatz und klicken sie dann auf Messwert erfassen, um die Probe einzulesen.

Name: Testreihe 2018-08-13 19:12  

Leerproben-Messtriple

1. Messung: 0.25 V X

2. Messung: 0.25 V X

3. Messung: 0.25 V X

Gemittelte Spannung der Leerprobe: 0.25 V

Leerprobenmessung abschließen

Leerprobe erfassen

An dieser Stelle wird ein Leerproben-Messtriple erfasst. Bitte geben Sie eine Leerprobe ohne Malondialdehyd in den Messplatz und klicken sie dann auf Messwert erfassen, um die Probe einzulesen.

Nach dem Erfassen einer neuen Leerprobe leitet das Programm zurück auf die Messübersicht. Im folgenden Beispiel wurde eine weitere Messung mit neuer Leerprobe durchgeführt.

Kalibrierungswerte und erste Leerprobe werden in der Tabelle in blau, neue Leerproben-erfassungen türkis und Messungen in weiß hinterlegt. Sofern mit einem neu eingelesenen Leerprobenabzug gearbeitet wird, erscheint eine entsprechende Hinweismeldung. Messungen mit verändertem Leerprobenabzug erscheinen als grüne Punkte im Graphen der Abbildung, befinden sich jedoch aufgrund des geänderten Achsenabschnittes nicht auf der Regressionsgeraden.

Name: Testreihe
2018-08-13 19:13

Spannung/Konzentration MDA, Korrelationskoeff.: 0.99969

f(x)=0.001021x [V/nM] +0.2224 [V]
 ↓ Messung
 ▲ Kalibrierwert

Leerprobe:

Eine neue Leerprobe wurde nach der Kalibrierung eingelesen. Sie findet für diese Messung statt der ursprünglichen Verwendung. Leerprobenwert: 0.25 V

Conz [nM]	Ø-V [V]	V1 [V]	V2 [V]	V3 [V]	Datum	Name	Kommentar	DB-ID	Leerprobenabzug
0	0.2167	0.21	0.22	0.22	2018-08-13 17:57	cal	Leerprobe	0	0
100	0.3267	0.32	0.32	0.34	2018-08-13 17:57	cal	Kalibriermesswert	1	0.2224
200	0.4333	0.44	0.43	0.43	2018-08-13 17:57	cal	Kalibriermesswert	2	0.2224
500	0.73	0.73	0.73	0.73	2018-08-13 17:57	cal	Kalibriermesswert	3	0.2224
247.6983	0.4753	0.478	0.474	0.474	2018-08-13 17:57	Testpatient		4	0.2224
0	0.25	0.25	0.25	0.25	2018-08-13 19:12	blank	Leerprobe	5	0
223.3105	0.478	0.478	0.478	0.478	2018-08-13 19:12	Testpatient2		6	0.25

Leerprobe neu bestimmen

zur Messung

Nachdem die Messreihe angelegt wurde, erscheint diese in der Messreihenansicht. Die jeweils aktuelle Messreihe ist blau hinterlegt. Diese kann nicht gelöscht werden. Ältere Messreihen erscheinen in weiß. Sofern eine alte Messreihe aufgerufen wird, erscheint in der Messübersicht eine Warnmeldung darüber, dass es nicht empfohlen ist, eine weitere Messung innerhalb einer alten Messreihe durchzuführen.



Weitere Informationen: Löschen von Messungen

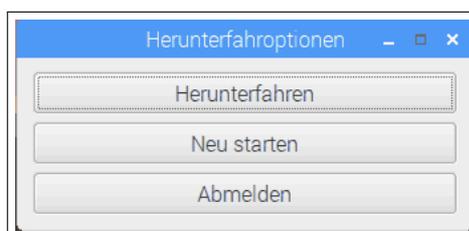
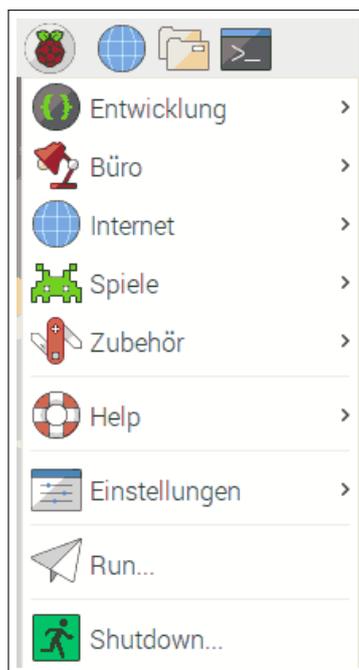
Während die Möglichkeit zur Löschung (alter) Messreihen implementiert wurde, ist das Löschen einzelner Messungen nicht möglich. Hierfür gibt es zweierlei Gründe:

Zum Einen erfordert die Funktion einige technische Umbauten am Programm, insbesondere im Hinblick auf die Verwaltung eindeutiger IDs, die jeder Messung zugewiesen sind. Darüber hinaus ist die Implementierung von Ausnahmeregelungen erforderlich, um zu verhindern, dass Kalibrierungsmessungen entfernt werden können. Zum Anderen bedeutet das Löschen von Messungen, dass die Messkontinuität verloren geht und eine spätere Nachvollziehbarkeit von Messungen u.U. nicht mehr gegeben ist.

Demgegenüber standen Argumente, das Löschen zu ermöglichen. Hierzu gehört das Löschen offensichtlicher Fehl-Messungen, die erst nach dem Speichervorgang offenkundig werden, eine eventuelle gesetzliche Verpflichtung zur Löschung nach europäischem Datenschutzrecht, sowie dem Grundsatz, dem Nutzer im Zweifel einen größtmöglichen Entscheidungsspielraum zu geben, solange keine anderen Belange (z.B. rechtliche oder technische) dagegen sprechen.

5.4 Ausschalten des Gerätes

Klicken sie auf das Startmenü oben links in der Ecke der Startleiste. Klicken sie auf „Shutdown“ und im anschließenden Dialog auf „herunterfahren“. Nach wenigen Sekunden ist das Gerät heruntergefahren und sie haben die Möglichkeit, das Gerät vom Netz zu trennen.



6 Realisierung der Software

Um dem sequentiellen Leser dieser Thesis zuerst die Oberfläche mit Hilfe der Anleitung näher zu bringen, erfolgt erst an recht später Stelle, genau hier, die Beschreibung des Quellcodes.

Eine erste Test-Implementierung erfolgte – analog zur Spezialisierung – mit Python 2.7, sowie dem Python-Flask Webserver für die Programmierung des Webinterface. Die Grundlagen boten die Tutorials unter [17] und [18]. Hierbei stellte sich heraus, dass die Programmierung einer modernen Oberfläche mit Flask nicht zum gewünschten Erfolg führen würde. Nach Gesprächen mit Webentwicklern wurde das Konzept umgestaltet und das System in ein Frontend- und Backend-System aufgeteilt. Das Frontend-System stellt die graphische Oberfläche bereit, das Backend-System übernimmt die Ansteuerung der Peripherie. In diesem Zusammenhang wurde das Backend auf die Python-Version 3.5 bzw. 3.6 portiert⁹.

6.1 Backend

Der Backend-Server ist sehr schlank und dient zur Kommunikation mit dem Frontend über Websockets, sowie zur Verwaltung der Peripherie-Schnittstellen.

Die Hauptanwendung des Backends besteht aus dem Python-Programm `app.py` und weiteren, aufgabenspezifischen Modulen, die im Folgenden benannt werden. Darüber hinaus dient die Zusatzanwendung `CreateEmptyMainDB.py` dazu, eine leere Datenbankdatei zu erstellen, sofern die vorhandene Datei entfernt wurde.

- `app.py`: Kommuniziert über Websockets mit dem Frontend, importiert Funktionen:
 - `dbTools.py`: Verwaltet Datenbank Lese- und Schreibzugriffe
 - `drawTools.py`: Berechnung und Zeichnung der Regressionsgeraden
 - `mcpTools.py`: Ansteuern des AD-Wandlers
 - `exportTools.py`: Export von Graphik und Messwert-Tabelle auf USB-Stick
- `CreateEmptyMainDB`: Nach manuellem Löschen der Datenbank-Datei erstellt dieses Programm eine neue Datenbank mit leerer Haupttabelle.

Eine neue Hauptdatenbank kann wie folgt über ein Terminal angelegt werden. Achtung: Hierfür werden alle bisherigen Messdaten gelöscht.

Anlegen einer neuen Hauptdatenbank

```
cd ADpiServer
rm database/main.db
python3 CreateEmptyMainDB
```

⁹Auf dem Ubuntu-Entwicklungssystem läuft Python 3.6, auf dem Raspbian System des Raspberry Pi kommt Python 3.5 zum Einsatz

Der Quelltext ist an wichtigen Stellen kommentiert. An dieser Stelle werden ein paar Stichpunkte zu einigen wichtigen Programmfunktionen des Backends genannt:

Berechnung der Regressionsgeraden Für die Regressionsgerade wird die in Python-Numpy enthaltene Polyfit-Funktion verwendet. Die Funktion liefert die Geradenparameter (Achsenabschnitt und Steigung) zurück, welche von Matplotlib zur Zeichnung einer Geraden verwendet werden. Darüber hinaus wird mit der Python-Numpy Funktion „corrcoef“ der Korrelationskoeffizient berechnet und in den Titel der Abbildung eingefügt. Die daraus resultierende Graphik wird als PNG-Datei an das Frontend gesendet. Die Umsetzung zur Erzeugung eines PNG Bildes zur Anzeige via Flask stammt von [19].

Export Die Export-Funktion sucht nach Mount-Points im Verzeichnis /media/pi, in welchem Raspbian Datenträger einhängt. Wird genau ein Mount-Point gefunden, wird die Abbildung als PNG und eine CSV-Datei mit dem Tabelleninhalt der Messtabelle mitsamt passender Spaltenbeschriftungen abgespeichert. Im Anschluss wird der eingehängte Datenträger ausgehängt und eine entsprechende Meldung an das Frontend übergeben.

Messwert erfassen Die Abfrage des AD-Wandlers erfolgt mit der GPIO-Zero-Bibliothek. Der AD-Wandler wird fünf mal mit einer Pause von je 50 ms abgefragt. Der jeweils höchste und niedrigste dieser fünf Messwerte wird verworfen und über die verbliebenen drei Werte ein arithmetisches Mittel gebildet. Die GPIO-Zero-Bibliothek liefert Werte zwischen 0 und 1 zurück. Um das Ergebnis in eine Spannung zu konvertieren, wird mit der Referenzspannung von 2,5 V multipliziert. Im Anschluss erfolgt eine Rundung auf die 5. Nachkommastelle, da ein 12 Bit AD-Wandler lediglich 4096 Werte unterscheiden kann.

6.2 Datenbankkonzept

Die Datenbank ist mit Sqlite realisiert und besteht aus einer Haupttabelle („main“) und je einer Messtabelle pro Messreihe. Als Informationsgrundlage für die Nutzung der Sqlite-Schnittstelle unter Python diene [20], sowie insbesondere in Bezug auf manipulationssichere Insert-Befehle die Informationen von [21]

6.2.1 Haupttabelle main

Die Haupttabelle enthält folgende Spalten:

- id (integer Primary Key): Eindeutige ID der Messreihe
- name (text): Name der Messreihe
- time (date): Erstellungsdatum der Messreihe
- funcX (real): Steigung des Graphen
- funcB (real): Achsenabschnitt des Graphen

6.2.2 Messtabelle measure

Die Messtabellen haben folgendes Namensschema:

„measure“ + ID der Messreihe der Haupttabelle, z.B. „measure1“, „measure2“, etc.

Die Messtabelle enthält folgende Spalten:

- id (integer Primary Key): Eindeutige ID der Messung
- voltage1 (real): Spannungswert 1 des Messtripples
- voltage2 (real): Spannungswert 2 des Messtripples
- voltage3 (real): Spannungswert 3 des Messtripples
- voltage0 (real): Durchschnittliches, arithmetisches Mittel der drei gemessenen Spannungswerte. Wird im Frontend berechnet.
- concentration (Integer): Konzentration der Malondialdehydprobe. Wird bei der Kalibrierung vom Benutzer festgelegt und bei Messungen im Frontend berechnet.
- date (date): Datum der Messwerterfassung
- name (text): Name des Messwertes: 'cal' bei initialer Leerprobenerfassung und Kalibrierung, 'blank' bei erneuter Leerprobenerfassung, ansonsten vom Benutzer festgelegt.
- comment (text): Kommentar zum Messwert: Wird bei Messung vom Benutzer festgelegt; optionales Feld.
- funcB (real): Verwendeter Leerprobenabzug für die Messung.

Eine Messung, d.h. eine Zeile der Tabelle, besteht immer aus einem Messtriple mit drei Einzelmessungen, über die ein arithmetisches Mittel gebildet wird.

Weitere Informationen: Erstellen und Löschen von Messreihen

Beim Erstellen einer neuen Messreihe wird eine neue Tabellenreihe in der „main“-Tabelle erzeugt. Hierfür wird die letzte verwendete Messreihen-ID aus der Haupttabelle ausgelesen und um eins inkrementiert. Sofern noch kein Tabelleneintrag existiert, wird „0“ zurückgeliefert und um eins inkrementiert, d.h. die erste Messreihen-ID lautet „1“. An einigen Stellen im Frontend wird überprüft, ob einer Messreihe bereits eine Messreihen-ID zugewiesen ist. So kann zwischen einer gerade im Erstellungsprozess befindlichen, nicht kalibrierten Messreihe und einer bereits vollständig kalibrierten Messreihe unterschieden werden. Erstere können beispielsweise nicht exportiert werden. Diese Vergleiche führen zu Problemen, sofern diese ID „0“ ist.

Sofern eine Messreihe gelöscht wird, wird die zugehörige Messtabelle gelöscht und die zugehörige Zeile aus der Haupttabelle entfernt. Die aktuelle Messreihe kann nicht gelöscht werden. Dies wurde eingeführt, um eine versehentliche Löschung der aktuellen Messreihe – trotz Sicherheitsabfrage – zu verhindern, da es in der Regel keine Notwendigkeit zur Löschung der aktuellen Messreihe gibt. Zur Löschung der aktuellen Messreihe muss im Vorfeld eine neue Messreihe erstellt werden, dann kann die vormals aktuelle Messreihe gelöscht werden. Dies führt dazu, dass Messreihen-IDs nicht erneut vergeben werden. In der Praxis gab es keine Probleme mit nach Löschung erneut vergebenen Messreihen-IDs, aber auch an dieser Stelle könnte das Vorgehen Fehlern vorbeugen. Der Teufel ist bekanntlich ein Eichhörnchen.

6.3 Frontend

Das Frontend besteht aus einer Vorlage des Vue-Framework[12] für den Node-Webserver. Die Vorlage besteht aus Konfigurationsdateien für Vue sowie Erweiterungen, z.B. Export- und Papierkorb-Symbole von FontAwesome sowie Bulma als CSS-Framework[22], welches z.B. Button-Elemente und Tabellen mit vorgefertigten Design-Elementen verschönert.

Vue bietet die Möglichkeit, innerhalb der HTML-Seite Logik-Elemente zu integrieren, sodass HTML-Tabellen dynamisch erzeugt werden können, indem mit einer for-Schleife über ein Array aus Messwerten iteriert wird. Mit if-Abfragen können in Abhängigkeit interner Programmzustände Seitenelemente ein- oder ausgeblendet werden. Auf diese Weise werden die vier Zustände (Leerprobe, Kalibrierung, Messübersicht, Messung) des Messvorgangs realisiert. Darüber hinaus können Elemente auch in Abhängigkeit von internen Variablen verschiedene Eigenschaften, z.B. Farben, erhalten. Diese Funktion wird z.B. zur Farbgebung der Messtabelle benutzt, um Kalibrierungswerte und Leerproben-erfassungen gesondert einzufärben. Auch in der Messreihenansicht wird die aktuellste Messreihe auf diese Weise eingefärbt.

Der Quellcode beinhaltet HTML5 Webseitenlayout, Javascript Funktionen für die Implementierung von Logik-Routinen, sowie untergeordnet auch CSS-Code für graphisches Design. Ein erheblicher Teil der Anwendungslogik ist in Javascript-Funktionen des Frontend-Systems programmiert. Die HTML5 Elemente beinhalten – wie oben beschrieben – in geringem Umfang ebenfalls Ablauf-Logik, die das Vue Framework zur Verfügung stellt.

Das Hauptaugenmerk der hier beschriebenen Dokumentation liegt auf dem HTML5 und Javascript Quellcode, welcher sich hauptsächlich auf drei Unterseiten verteilt.

6.3.1 layouts/default.vue

Die Layout-Datei ist dafür konzipiert, webseitenübergreifende Seitenelemente zur Verfügung zu stellen. Hierzu gehören Kopf- und Fußzeilen oder Menüstrukturen. Sie enthält die blaue Menüzeile am oberen Rand, welche ursprünglich als Navigationsmenü ange-dacht war. Nachdem das Design des Programmes in mehreren Schritten in Bezug auf die Oberfläche stets vereinfacht wurde, verblieb am Ende nur die Titelzeile mit der Möglichkeit durch ein Klick auf den Text zurück in die Messreihenansicht zu gelangen.

6.3.2 pages/index.vue

Die Index-Seite wird beim Programmstart gezeigt und enthält die Messreihenansicht. Aus dieser heraus kann eine neue Messreihe erstellt werden, oder eine bestehende Messreihe geöffnet werden. Die Liste der Messreihen wird beim Seitenaufbau aus der Datenbank abgerufen.

6.3.3 pages/calibrate.vue

Diese Seite enthält den gesamten Messprozess. Dieser ist durch mehrere Zustände definiert. Die obere Leiste mit dem Namen der Messreihe und der Export- und Löschfunktion wird in jedem Zustand zwar angezeigt, einzelne Funktionen sind jedoch nicht in jedem Zustand verfügbar, sondern deaktiviert. So ist der Export und die Möglichkeit zum Löschen einer Messreihe nur in der Messübersicht möglich. Der Name wird in jedem Zustand angezeigt, änderbar ist er hingegen nur in der initialen Leerprobenerfassung und der Kalibrierung.

Leerprobenerfassung Es werden zwei Zustände unterschieden. Zum einen geht die Leerprobenerfassung jeder Kalibrierung voraus. Zum anderen kann eine erneute Leerprobenerfassung vor einer Messung erfolgen. Beim Beenden der Erfassung werden diese Zustände unterschiedlich behandelt. Im ersten Fall wird die Leerprobenerfassung in die Kalibrierung überführt, im zweiten Fall wird die Leerprobe wie eine Messung behandelt und gespeichert, und der Leerprobenabzug temporär ersetzt, jedoch nicht als solcher dauerhaft in die Datenbank geschrieben.

Kalibrierung In diesem Zustand wird intern ein Array¹⁰ angelegt, in welchem Messwerte gespeichert werden. Der Benutzer kann das interne Array beliebig durch drücken auf „weitere Referenzprobe“ verlängern. Es wird vor Abschluss der Kalibrierung geprüft, ob für jede Referenzprobe jede der drei Messungen erfasst wurden, eine gültige Konzentrationsangabe gemacht wurde und spätestens hier, wenn nicht bereits in der Leerprobe, ein Messreihenname vergeben wurde. Bei Abschluss der Kalibrierung wird das Mess-Array in die Datenbank geschrieben und in die Messübersicht gewechselt.

Messübersicht Die Messübersicht wird nach der Kalibrierung, der Messung sowie der erneuten Erfassung einer Leerprobe aufgerufen. Der Wechsel in diesen Zustand initiiert die Berechnung der Regressionsgeraden und der Abbildung eben dieser. Darüber hinaus ist die Messübersicht der Zustand, der beim Laden einer bestehenden Messreihe aufgerufen wird. Hierfür werden alle Variablen so gesetzt, wie sie auch nach Abschluss einer Kalibrierung gesetzt sind. Das Mess-Array wird hierfür aus der Datenbank geladen. Die Abbildung wird nicht in der Datenbank gespeichert, sondern individuell durch das Backend berechnet. Das ist insofern sinnvoll, als die Berechnung recht zügig vonstatten geht und auch jeweils aktuelle Messwerte in der Graphik angezeigt werden können.

Messung Diese ist ähnlich aufgebaut wie die Kalibrierung. Im Unterschied zu dieser wird hier selbst ein Name für die Messung vergeben und stattdessen die Konzentration aus der Regressionsgeraden berechnet. Nach Abschluss der Messung werden die neu generierten Daten in der Datenbank gespeichert und es wird in die Messübersicht gewechselt.

¹⁰Das Array ist ein zweidimensionales Array, d.h. es enthält beliebig viele Elemente (Messungen) in der ersten Dimension. In der zweiten Dimension ist es ein Abbild der 10-spaltigen Messtabelle aus der Datenbank.

Achtung: Sicherheit bei der Webprogrammierung

An dieser Stelle ein Hinweis auf die Programmsicherheit:

Der Autor dieser Thesis ist kein ausgebildeter Programmierer. Viele Merkmale sicheren Programmierens sind dem Autor bekannt, es wird dennoch darauf hingewiesen, dass Sicherheitslücken in dem Programm vorhanden sein können.

Sowohl Python als auch Javascript beherrschen die automatische Konvertierung von Variablentypen, sodass z.B. Vergleiche von Zahlen (Integer) mit Text, in dem Zahlen als String gespeichert sind, möglich werden. Typübergreifende Vergleiche von Variableninhalten führen jedoch in manchen Fällen zu unerwarteten Ergebnissen, sodass üblicherweise von einem solchen Vergleich abgeraten wird. Daher wurde bei der Programmierung Wert darauf gelegt, dass typübergreifende Vergleiche möglichst vermieden werden und stattdessen eine explizite Typkonvertierung durchgeführt wird.

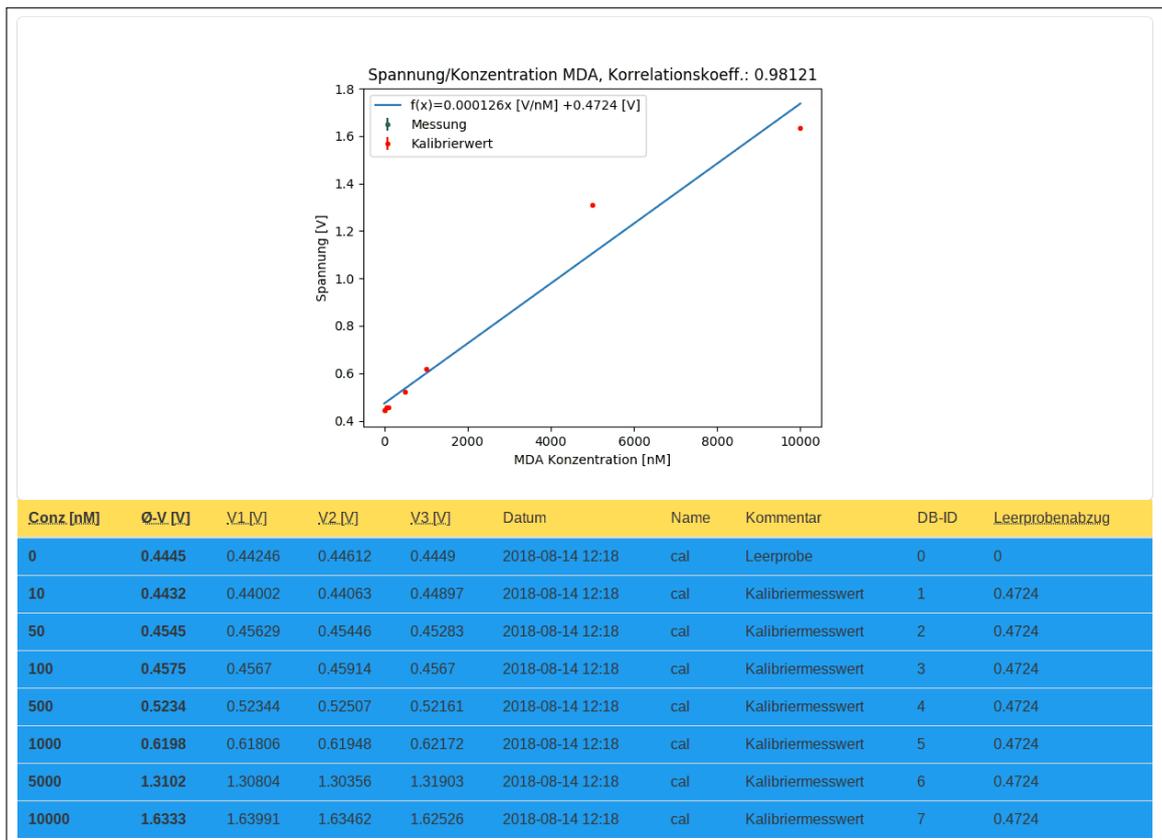
Vom Benutzer im Frontend eingegebene Werte sind grundsätzlich vor dem Weiterverarbeiten zu überprüfen. Ein Benutzer könnte versuchen, in Texteingabefelder zur Übergabe von Namen, Kommentaren oder Konzentrationsangaben stattdessen Datenbank-Befehle einzugeben und z.B. die Löschung von Datenbankentabellen zu initiieren. Um eine solche „SQL-Injection“ zu verhindern, muss sichergestellt werden, dass keine Datenbank-Befehle über Eingabefelder eingegeben oder auf andere Art und Weise vom Benutzer übermittelt werden können. Hierfür wurden die von der Python-Datenbankschnittstelle bereitgestellten Funktionsaufrufe genutzt, die die vom Benutzer übergebenen Parameter bereinigen („sanitizing“).

Bei der Übergabe zwischen Frontend und Backend besteht jedoch ein Vertrauensverhältnis, welches als Angriffsvektor genutzt werden kann. Daten, die vom Frontend zum Backend übertragen werden, werden an vielen Stellen nur unzureichend kontrolliert, sodass durch gezielte Manipulation der Daten im Frontend Angriffe auf das Backend möglich sind. Üblicherweise sind alle vom Frontend gemeldeten Werte grundsätzlich zu überprüfen, i.d.R. erst vom Frontend, um den Benutzer bei Fehleingaben direkt informieren zu können, und im Anschluss erneut vom Backend, um bei Manipulation des Frontends Angriffsversuche zu unterbinden. Dieses Vorgehen ist de facto Standard bei der Programmierung von Seiten, die einer breiten Öffentlichkeit im Internet zur Verfügung stehen. Dies ist mit einem erheblichen Mehraufwand verbunden, führt zu einer großen Anzahl an Code-Dopplungen und auch zu einem erhöhten Bedarf an Rechenleistung, die auf dem Raspberry Pi grundsätzlich knapp bemessen ist. Daher, und aufgrund des erheblichen Mehraufwandes und fehlender Erfahrung wurde auf solche Sicherheitsmaßnahmen verzichtet. In Anbetracht der ausschließlich lokalen Nutzung des Gerätes ist dieses Vorgehen vertretbar. Es ist aber nicht ratsam, das Gerät in öffentlichen Netzen zu betreiben.

7 Test-Messung

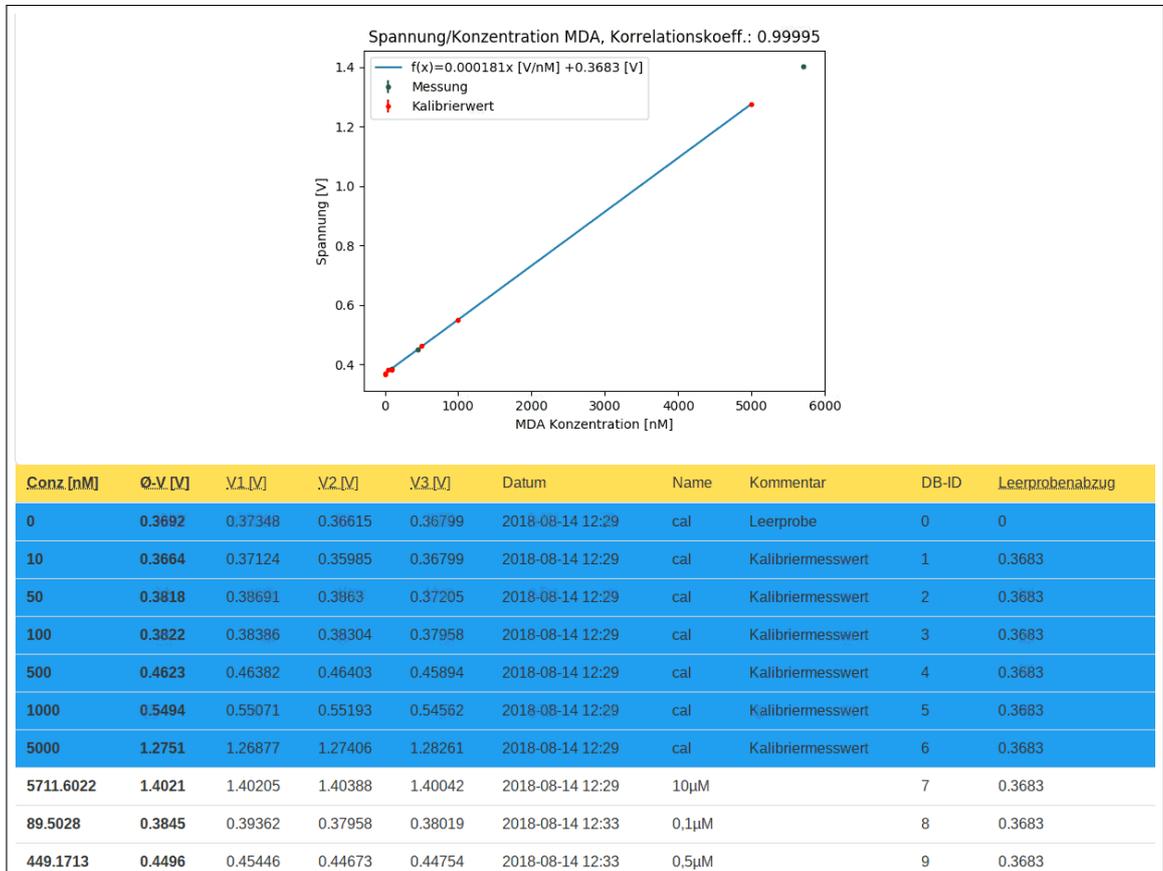
In diesem Kapitel wird das Ergebnis einer Test-Kalibrierung und -Messung gezeigt. Es wurde jeweils zuerst eine Kalibrierung durchgeführt, beim zweiten Test wurden im Nachgang Messungen durchgeführt. Die Messungen wurden nach ihren Konzentrationen benannt.

Die erste Test-Kalibrierung erfolgte ca. 20 min nach Probenerstellung im Labor. Für die Kalibrierung wurden Werte bis 10.000 nM herangezogen. Messungen wurden nach der Kalibrierung nicht durchgeführt.



Im Ergebnis der Messung zeigt sich, dass Proben über 5000 nM im Sättigungsbereich liegen, d.h. hier besteht kein linearer Zusammenhang zwischen Probenkonzentration und Spannung mehr. Eine Kalibrierung sollte daher höchstens Werte mit 5000 nM berücksichtigen.

Die Messung wurde daher mit den gleichen Proben zusammen mit Herr Schumacher ca. 40 min nach Probenerstellung wiederholt, ohne das Proben mit 10.000 nM für die Kalibrierung herangezogen wurden.



Hier ist erkennbar, dass die Leerprobenspannung bei verwendung identischer Proben deutlich abgenommen hat. Die Leerprobenspannung unterliegt demnach zeitlichen Änderungen, die bei der Messung und Kalibrierung berücksichtigt werden muss. Hier ist im Folgenden zu überprüfen, inwiefern sich dies auf die Messgenauigkeit auswirkt. Gegebenenfalls muss gewährleistet sein, dass das Alter der Proben für die Bestimmung der Malondialdehyd-Konzentration identisch ist.

8 Fazit

8.1 Zusammenfassung

Das fertiggestellte Datenbanksystem des Fluoreszenzphotometers erfüllt alle an diese Arbeit gestellten Anforderungen. Es ist kompakt, bietet eine intuitive Benutzeroberfläche und speichert alle Daten in einer Datenbank, die auf Wunsch exportiert werden kann.

Darüber hinaus wurde das Gehäuse des Fluoreszenzphotometers fertiggestellt. Es verfügt nun über einen Probendeckel, sodass ein Wechsel der Probe schnell möglich ist, ohne das Gehäuse zu öffnen. Darüber hinaus wurde die Höhe des Gehäuses deutlich reduziert, um einen Wechsel der Probe über den Probendeckel zu ermöglichen. Als positiver Nebeneffekt hat sich das Gewicht des Gehäuses merklich reduziert.

Der Raspberry Pi wurde so in das Gehäuse integriert, dass die Anschlussbuchsen für Strom und Bildschirm sowie die USB-Ports von außen erreichbar sind. Das ist für einen einfachen Anschluss von Monitor und Eingabegeräten sinnvoll, ebenfalls zum Einstecken eines USB-Sticks für den Datenexport. Die Messgenauigkeit liegt gleichauf mit der des Multimeters, d.h. die Messergebnisse entsprechen den Tests aus der Arbeit von Herrn Joppien.

Alle Arbeitsschritte zur Reproduktion des Gerätes können der Anleitung von Herr Joppien und dieser Thesis entnommen werden, sodass ein Nachbau in Kleinserie möglich ist.

8.2 Ausblick

Mit weiterer Optimierung und Standardisierung des Gehäuses, beispielsweise durch Designvorlagen aus einem 3D-Drucker, könnte in Zukunft das Gewicht sowie die Kosten der Aparatur reduziert werden. Auch eine regelmäßige Neukalibrierung könnte durch diese Standardisierungsmaßnahmen und das Entfernen beweglicher Schraubverbindungen entfallen und die Praxistauglichkeit dadurch erhöhen. Erstrebenswert ist auch der Tausch des Lasers gegen eine LED, um die streulichtintensive Lochspiegelkonstruktion zu ersetzen.

Bei den Test-Messungen zeigte sich eine deutliche, zeitliche Abhängigkeit der Leerprobenspannung. Diese muss mit dem Ziel untersucht werden, einheitliche Testbedingungen zu schaffen, um Kalibrierungen zur Verbesserung der Praxistauglichkeit zu reduzieren.

A Quellcode

Aus praktischen Gründen wird der Quellcode des ADpi-Programmes an dieser Stelle nicht abgedruckt. Er befindet sich auf der beiliegenden CD. Sofern sie in diesem Moment eine PDF-Datei lesen, ist der Quellcode als 7zip-Archiv im Anhang dieser PDF-Datei zu finden. Im Adobe-Reader haben sie unten links über die Büroklammer die Möglichkeit den Anhang zu speichern. Unter Umständen verhindert der Adobe Reader das Abspeichern des Anhangs im standardmäßig aktiven „Protected Mode“. Sie können den Quellcode alternativ mit PDFTK extrahieren.

PDFTK Anhang extrahieren

```
pdftk PDFFILE.pdf unpack_files output OUTPUTDIR/
```

B Abkürzungsverzeichnis

Board-Notation siehe BCM-Notation

BCM-Notation Für die Angabe der GPIO Pin Belegung existieren zwei Möglichkeiten: Die Durchnummerierung der Kontakte abgezählt auf der Raspberry Pi Platine (Board-Notation). Diese Notation zählt auch Kontakte, welche keine GPIO Funktionalität besitzen, darunter 5V, 3,3V und GND Kontakte. Alternativ existiert die im verbauten Broadcom System-on-a-Chip (SOC) intern verwendete Belegung (BCM-Notation). Die verwendete Notation ist stets angegeben.

CSS Cascading Style Sheets ist eine Beschreibungssprache für Webanwendungen, die die graphische Gestaltung und Aufbereitung übernimmt. Zusammen mit Javascript für die Integration von Logik-Funktionen und HTML (Hypertext Markup Language) für die Beschreibung des Inhalts bilden sie die Kernsprachen des World Wide Webs.

CSV Comma-separated values: Eine Textdatei, die Tabelleninformationen enthält. Zeilen des Dokuments entsprechen den Zeilen der Tabelle. Tabellenespalten werden im Textdokument durch einen speziellen Zeichentrenner kodiert. Dieser ist variabel; meist handelt es sich jedoch um Semikolons, Tabulatoren oder das namensgebende Komma.

DVI Digital Visual Interface: Schnittstelle zum Anschluss digitaler Anzeigegeräte.

GPIO General Purpose Input Output: Kontaktstift, dessen Verhalten als Ein- oder Ausgangskanal frei programmierbar ist. Als Eingabekanal kann er detektieren, ob eine Spannung anliegt. Als Ausgabekanal kann er je nach Programmierung eine Spannung ausgeben – oder nicht. So können angeschlossene Geräte (LEDs, Relais) gesteuert werden.

HTML siehe CSS

I2C Bus Inter-Integrated Circuit: Standard für seriellen Datenbus, der zur Kommunikation zwischen integrierten Schaltungen genutzt wird.

NAS Network Attached Storage: Festplattenspeicher, welcher – im Gegensatz zu klassischen USB-Festplatten – über ein Netzwerkprotokoll verfügbar gemacht wird. NAS-Systeme sind zur Umsetzung des Netzwerkprotokolls immer als Mini-Computer realisiert, d.h. sie besitzen einen Hauptprozessor und Speicher. Nicht selten kommt Linux als Betriebssystem zum Einsatz. Hierdurch lassen sich einige Systeme in Bezug auf die unterstützten Protokolle und Dienste erweitern.

NTP Network Time Protocol: Internetprotokoll zur Synchronisation der aktuellen Uhrzeit mit einem Zeitserver.

VPN Virtual Private Network: VPN bezeichnet das Verbinden entfernter Standorte mit einem virtuellen Netzwerk über das Internet, um Netzwerke dieser Standorte zu einem virtuellen, lokalen Netz zusammenzuschließen.

Repository Auf Linux und BSD basierende Betriebssysteme nutzen zur Installation von Softwareprogrammen überwiegend die in der Windows-Welt unbekannteren Paket-Verwaltungssysteme (Software-Repositories). Hierbei handelt es sich um digitale Softwarearchive. Die im Archiv enthaltenen Programme und Bibliotheken werden regelmäßig aktualisiert und können mit Kommandozeilenbefehlen oder graphischen Programmen automatisiert installiert oder wieder entfernt werden.

SMD Surface-Mount Device ist eine Bauteilelement-Form, bei der die Bauteile mittels lötfähiger Anschlussflächen auf die Oberfläche einer Leiterplatte gelötet werden, entgegen dem bedrahteten Gehäusertyp, welcher auf der dem Bauteil gegenüberliegenden Seite verlötet wird.

SPI Bus Serial Peripheral Interface: Standard für synchronen, seriellen Datenbus, mit dem Geräte nach dem Master-Slave-Prinzip verbunden werden können.

TO92 ist ein weitverbreiteter 3-Bein Gehäusertyp, der (aufgrund der 3 Beine) meist für Transistoren verwendet wird. Es handelt sich um ein bedrahtetes Gehäuse.

Z-Diode Eine Z-Diode ist eine Diode, die zum dauerhaften Betrieb in Sperrrichtung oberhalb der Durchbruchspannung ausgelegt ist. Kennzeichnend für den Einsatz in diesem Projekt ist der konstante Spannungsabfall bei Betrieb in Sperrrichtung, weshalb sie als Referenzspannungsquelle eingesetzt wird.

Literatur

- [1] C.Müller, K. Bartkowski, T.Hohlfeld: Thrombozytäre Freisetzung von Malondialdehyd als Index der Thromboxansynthese (2018)
- [2] T.Hohlfeld, N. Zimmermann, et al.: Pyrazolinone analgesics prevent the antiplatelet effect of aspirin and preserve human platelet thromboxane synthesis (2007)
- [3] Yannick Joppien: Bau eines Fluoreszenzphotometers zum Nachweis von Malondialdehyd (2016)
- [4] D.Schölzel: Entwicklung eines opto-elektronischen Aufbaus zur Bestimmung der Malondialdehydkonzentration in Blutplasma (2014)
- [5] Christian Krause: Optimierung eines computergestuetzten Temperaturreglers zur thermischen Aufbereitung einer medizinischen Probe (2017)
- [6] Raspberry Pi GPIO Pinout <https://pinout.xyz>
- [7] MCP3008: Am Raspberry Pi analoge Signale auslesen <https://tutorials-raspberrypi.de/raspberry-pi-mcp3008-analoge-signale-auslesen>
- [8] Spannungsmessung mit AD-Wandler am Raspberry Pi <https://www.direcs.de/2017/03/spannungsmessung-mit-ad-wandler-am-raspberry-pi>
- [9] MCP3204 Datenblatt <https://www.mouser.de/datasheet/2/268/21298c-68472.pdf>
- [10] mikrocontroller.net - Diskussion über Referenzspannungsquellen <https://www.mikrocontroller.net/topic/372268>
- [11] Raspberry Pi Geek: Echtzeituhr-Modul DS3231 sorgt für genaue Zeitangaben <http://www.raspberry-pi-geek.de/Magazin/2015/03/Echtzeituhr-Modul-DS3231-sorgt-fuer-genaue-Zeitangaben>
- [12] Webframework Vue.js <https://vuejs.org>
- [13] Vergleich Angular, React, Vue <https://jaxenter.de/javascript-vue-js-react-angular-66246>
- [14] Datenblatt LM4040 <http://www.ti.com/lit/ds/symlink/lm4040-n.pdf>
- [15] Gpiozero - SPI Devices - ADC http://gpiozero.readthedocs.io/en/stable/api_spi.html#analog-to-digital-converters-adc
- [16] Raspbian: Debian für den Raspberry Pi <https://www.raspberrypi.org/downloads>
- [17] Raspberry Pi Web Server using Flask to Control GPIOs <https://randomnerdtutorials.com/raspberry-pi-web-server-using-flask-to-control-gpios>

- [18] The Flask Mega-Tutorial
<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>
- [19] FlaskPlotlib - PNG for Flask with Mathplotlib
<https://gist.github.com/wilsaj/862153>
- [20] Python und SQL https://www.python-kurs.eu/sql_python.php
- [21] sqlite3 — DB-API 2.0 interface for SQLite databases
<https://docs.python.org/3.5/library/sqlite3.html>
- [22] CSS-Framework Bulma <https://bulma.io>

Alle Internetquellen wurden zuletzt aufgerufen am 14. August 2018