

Universität Würzburg
Lehrstuhl für Physik und ihre Didaktik
Fakultät für Physik und Astronomie

**Schriftliche Hausarbeit
im Rahmen zur ersten Staatsprüfung
für das Lehramt an Realschulen**

**Überprüfung und Verbesserung eines
Low-Budget Rastertunnel Mikroskops
auf Arduino-Basis**

Eingereicht von
Sprenger, Peter

Abgabe: 09/2016

**Betreuer & Prüfer:
Dr. Stephan Lück**

Danksagung

An dieser Stelle möchte ich allen Danken, die mich im Rahmen dieser Arbeit begleitet haben.

Ganz besonders möchte ich Herrn Dr. Stepan Lück danken, der meine Arbeit durch seine fachliche und persönliche Unterstützung begleitet hat und selbst abends gerne einen Ratschlag bereithielt.

Des Weiteren bedanke ich mich bei Herrn Jonas Dittmann, den mich in sein ehemaliges Projekt sehr detailliert eingeführt hat, sowie bei auftretenden Problemen gerne weiterhalf.

Außerdem bedanke ich mich bei den Mitarbeitern der Physikalischen Werkstatt der Universität Würzburg, welche bei der Umsetzung von Ideen beratend zur Seite standen.

Ferner möchte ich mich bei Pinar Özkaya, für das Korrekturlesen dieser Arbeit, bedanken.

Darüber hinaus möchte ich mich ganz besonders bei Seifeddin Mustafa und Viktor Giss bedanken, die mir während der hunderten Stunden Wartezeit auf Scanergebnisse, Gesellschaft leisteten und mit mir währenddessen über aktuelle Probleme meiner Arbeit diskutierten, wodurch stets Lösungsansätze entstanden.

Der größte Dank gilt meiner Frau Negra Sprenger, welche mich bei dieser Arbeit und vor allem bei der vorhergehenden praktischen Arbeit, stets moralisch unterstützt hat und immer für mich da war.

Inhaltsverzeichnis

1. Einführung	3
1.1. Anfangszustand	3
1.2. Zielsetzung	4
2. Funktionsweise des STM	5
2.1. Funktions-Prinzip der Messung (Tunneleffekt)	7
3. Aufbau & Elektronik	8
3.1. Schwingungsdämpfung der Scaneinheit	10
3.2. Funktionsweise des Vorverstärkers.....	10
3.3. Funktionsweise des Regelkreises (Z-Regelung)	12
3.4. Anpassung des Tunnelstromsignals und der X/Y-Steuerung an den Arduino Due .	15
3.5. Der Schrittmotor.....	19
3.6. Platine.....	21
4. Spitzenherstellung	23
5. Überprüfung der Platine	27
5.1. Überprüfen der X-Steuerung	28
5.2. Überprüfen der Y-Steuerung.....	31
5.3. Überprüfen der Nachregelung in Z-Richtung	33
6. Software	35
6.1. Arduino Due - Software	35
6.1.1. Spitzenannäherung/-entfernung	37
6.1.2. Messung	38
6.2. Computer-Software	40
6.2.1. Oberfläche	40
6.2.2. Kommunikation.....	43
6.2.3. Messung starten.....	44
6.2.4. Messung visualisieren	48
6.2.1. Speichern des Bildes/Ausdrucken und 3D-Ansicht	49
7. Nutzung des STM im Schülerlabor	50

7.1.	Allgemeine Erklärung	50
7.2.	Bedienungsanleitung zu selbstständigen Verwendung	52
8.	Abschließende Gedanken	55
9.	Literaturverzeichnis.....	56
10.	Abbildungsquellen.....	57
A	Anhang	58
B	Erklärung nach §29 LPO-I.....	59

1. Einführung

Diese Arbeit behandelt die Überprüfung und Verbesserung eines Low-Budget Raster-Tunnel-Mikroskops, kurz STM (aus engl. „Scanning tunneling microscope“), welches einige Jahre zuvor von Herrn Jonas Dittmann (vgl. (Dittmann, 2006)) nach einer Vorlage der TU Hannover, gebaut wurde. Einzelne Kapitel dieser Arbeit dienen lediglich zum Verständnis der anderen Kapitel und dürfen daher nicht als vollständige beziehungsweise vertiefte Inhalte angesehen werden.

Ein Rastertunnelmikroskop ermöglicht Aufnahmen von Oberflächen, in molekularer bis hin zu atomarer Auflösung, einer elektrisch leitenden Probe zu erstellen. Jedoch muss beachtet werden, dass nicht die wahre Oberfläche abgebildet wird, sondern vielmehr die elektrischen Eigenschaften des Moleküls bzw. des Atoms, das Bild beeinflussen (vgl. (Bai, 2000 S. 180)).

Der vorhandene Aufbau wurde im Physik - Schülerlabor verwendet, wies jedoch nach einiger Zeit einen Defekt auf. Dies gab den Anstoß das STM zu erneuern und mit neuer Technik zu versehen.

Die Wahl der Zulassungsarbeit fiel auf dieses Thema, da ich in meinem bisherigen Studienverlauf, nicht die Chance hatte, mit einem Rastertunnelmikroskop zu arbeiten. Des Weiteren sollte die Einarbeitung in die Materie mein Wissen im Bereich der Elektrotechnik, sowie Informatik erweitern. Die Auseinandersetzung mit diesem Projekt, ist für mich als angehende Lehrkraft eine Bereicherung, was mir erlaubt, zukünftige Projekte mit Schülern besser planen und durchführen zu können.

1.1. Anfangszustand

Wie oben schon beschrieben wurde, konnte das STM wegen mehreren Defekten, vor allem aufgrund von Platinendefekten (kalten Lötstellen und defekten Bauteilen), nichtmehr ohne Überarbeitung verwendet werden. Als Schnittstelle zwischen STM und Computer wurde ein Arduino Due eingesetzt, welcher mit einem, mit Python geschriebenen, Programm vom Computer gesteuert werden konnte. Die vom Arduino Due erfassten und an den Computer übermittelten Daten wurden anschließend mit einem, auf Matlab basierenden, Programm ausgewertet und als 2D Grafik visualisiert. Anschließend konnte diese Grafik mit dem Freeware Programm ImageJ in eine 3D Grafik gewandelt werden.

1.2. Zielsetzung

Die gestellten Ziele waren eine Überprüfung der Funktionstüchtigkeit des vorhandenen STM und die Überprüfung der Einzelkomponenten, sowie die Erstellung einer für Schüler einsetzbaren Betriebsanleitung.

Die selbstaufgelegten Ziele waren, wieder ein voll funktionsfähiges STM zu erhalten. Außerdem sollte ein neues Programm geschrieben werden, welches eine einfachere Handhabung, sowie eine einfachere Installation ermöglichte und keine weiteren Zusatzprogramme benötigte. Zusätzlich sollte eine automatische Annäherung der Spitze an die zu untersuchende Probe realisiert werden, ohne die Möglichkeit zu nehmen, den Annäherungsprozess von Hand vollziehen zu können, um bei technischen Problemen den Betrieb trotzdem zu gewährleisten, zumal eine Funktion des automatischen Betriebes zum damaligen Zeitpunkt noch nicht gesichert war.

In Bezug auf die Spitzen sollte ein Prozess entwickelt werden, der mit möglichst hoher Wahrscheinlichkeit eine brauchbare Spitze herstellt, also Spitzen die zu einem guten Scanresultat führen.

weit genähert, bis der Abstand ebenfalls dem Sollwert entspricht. Der Spannungswert der Nachregelung wird mittels Arduino Due nach jeder Positionsänderung, also an jedem Rasterpunkt, gemessen und an den Computer übertragen. Anhand dieser Messwerte und dem Wissen, in welcher Reihenfolge das Raster abgefahren wird, kann nun eine Topografie der Probe erstellt werden. Dabei ist die Größe der untersuchten Fläche abhängig von der Spannung, mit welcher der Piezosignalgeber angesteuert wird. Durch die Untersuchung eines DVD-Masters konnte jedoch die maximale Fläche sehr gut berechnet werden. „Der Abstand der Tracks beträgt bei der DVD $0,74 \mu\text{m}$ “ (Rehm, 2016).

Dadurch ergibt sich eine maximale Messfläche von ca. $2,7 * 0,74 \mu\text{m} \approx 2 \mu\text{m}$ (siehe Bild).

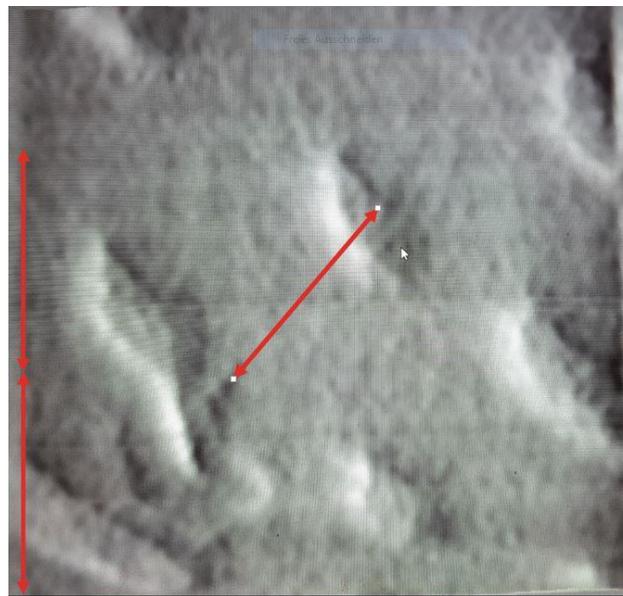


Abbildung 2: Aufnahme eines DVD-Masters mit dem STM-Eigenbau (19.10.2015)
Abstand der Tracks (Pfeil) $0,74 \mu\text{m}$

2.1. Funktions-Prinzip der Messung (Tunneleffekt)

Das Rastertunnelmikroskop funktioniert jedoch nur aufgrund des quantenmechanischen Phänomens des Tunnelns. Dabei beschreibt der Tunneleffekt das Ereignis, dass mit einer gewissen Wahrscheinlichkeit ein Teilchen, eine endliche Potentialbarriere überwinden kann, obwohl dessen Energie aus klassischer Sicht, dafür nicht genügen würde. Der Tunneleffekt ist dabei ein Resultat aus der Lösung der eindimensionalen Schrödinger-Gleichung für die Wellenfunktion Ψ . Die Wellenfunktion Ψ beschreibt dabei den quantenmechanischen Zustand eines Teilchens im Raum. Dabei ist sie, selbst innerhalb, sowie hinter der Potentialbarriere nicht Null, nimmt jedoch mit zunehmender Dicke dieser exponentiell ab.

Das Betragsquadrat $|\Psi|^2$ der Wellenfunktion beschreibt dabei, die Aufenthaltswahrscheinlichkeitsdichte eines Ortes, woraus folgt, dass selbst hinter einer Potentialbarriere, die Aufenthaltswahrscheinlichkeit eines Teilchens größer Null wird, obwohl dies aus klassischer Sicht nicht möglich ist. (vgl. (Demtröder, 2005))

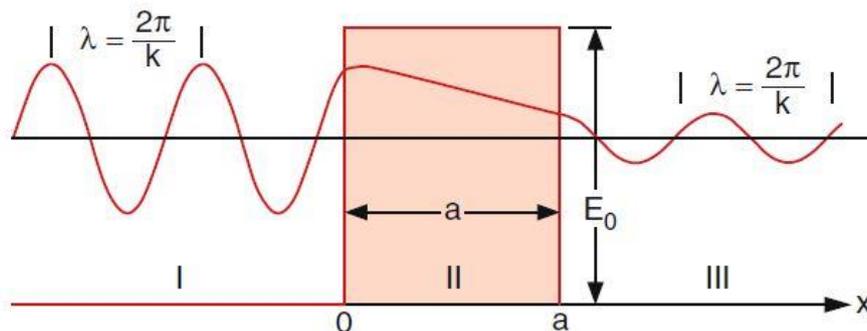


Abbildung 3: Tunneleffekt durch eine rechteckige Potentialbarriere
Quelle: Demtröder Experimentalphysik Band 3(2005), S. 119.

Daher können, bezogen auf ein STM, trotz keines Kontakts der Spitze mit der Probe, Elektronen von der Probe zur Spitze, beziehungsweise umgekehrt fließen. Dabei wirkt der Abstand zwischen den beiden Elektroden als Potentialbarriere, wodurch sich der sehr stark entfernungsabhängige Tunnelstrom, da mit linear steigendem Abstand die Tunnelwahrscheinlichkeit exponentiell abfällt, beschreibt.

3. Aufbau & Elektronik

Das Rastertunnelmikroskop kann in vier Bereiche unterteilt werden. Dies ist die Schwingungsdämpfung der Scaneinheit, die Scaneinheit selbst, in welche auch die zu untersuchende Probe eingespannt wird, die Platine welche zur Nachregelung der Messspitze und zum Transformieren der Mess- und Steuerspannung zuständig ist und als vierten Teil die Software inklusive Arduino Due als Kommunikationsschnittstelle zur Einstellung des Messvorgangs und Transformation der Messwerte in eine Topografie der untersuchten Probe.

Auf das Funktions-Prinzip der Scaneinheit wird in dieser Arbeit nicht vertieft eingegangen, da dies sehr ausführlich in der Facharbeit „Bau eines Rastertunnelmikroskops“ Kapitel 2.1 und Kapitel 2.2 von Jonas Dittmann behandelt wird, worauf diese Arbeit basiert (vgl. (Dittmann, 2006)).

Zum späteren Verständnis folgt nun jedoch eine kurze Erklärung der Scaneinheit.

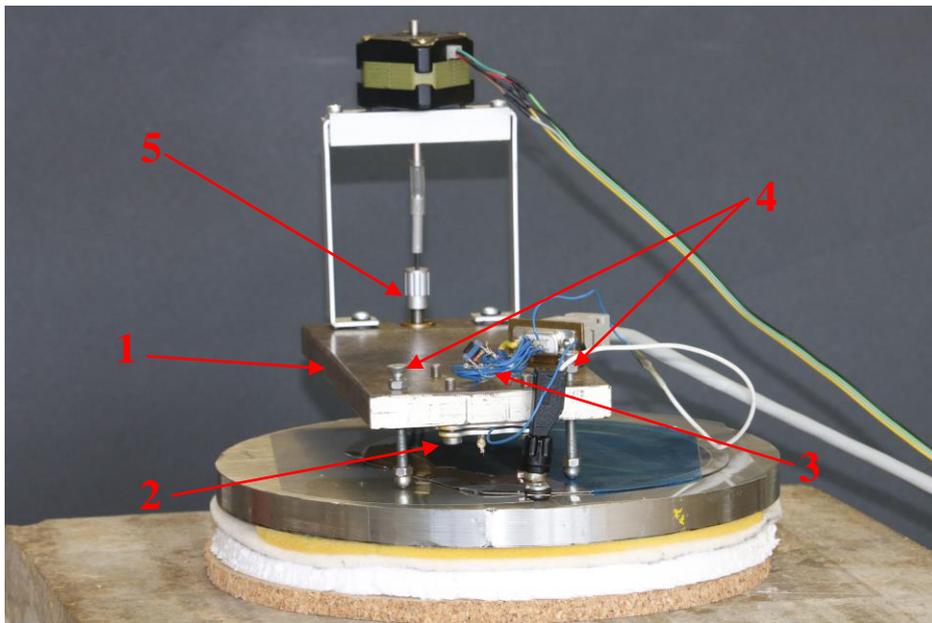


Abbildung 4: Scaneinheit mit Teil der Schwingungsdämpfungseinheit des STM

Wie auf Abbildung 4 zu erkennen ist, besteht die Scaneinheit aus einer Metallträgerplatte mit aufgeschraubtem Scankopf (Piezoeinheit (2) mit Vorverstärker (3)). Zur Annäherung des Scankopfes an die Probe ist die Metallplatte auf drei Schrauben gelagert, wobei die beiden Vorderen (4) Grobgewindeschrauben sind und daher zur Grobannäherung genutzt werden. Dagegen besitzt die hintere Schraube (5) eine Feingewinde und wird daher zur langsamen aber genauen Annäherung verwendet. Des Weiteren befindet sich auf der Oberseite (vorne), der Vorverstärker des Tunnelstroms (3), auf welchen in Kapitel 3.2 näher eingegangen wird. Auf der Unterseite des vorderen Teils der Platte, unterhalb des Vorverstärkers, befindet sich ein

Piezoschallgeber (2), der einseitig in 4 Teile geteilt ist, wodurch eine Steuerung der mittig eingefassten Halterung, mit eingeschraubter Messspitze, ermöglicht wird.

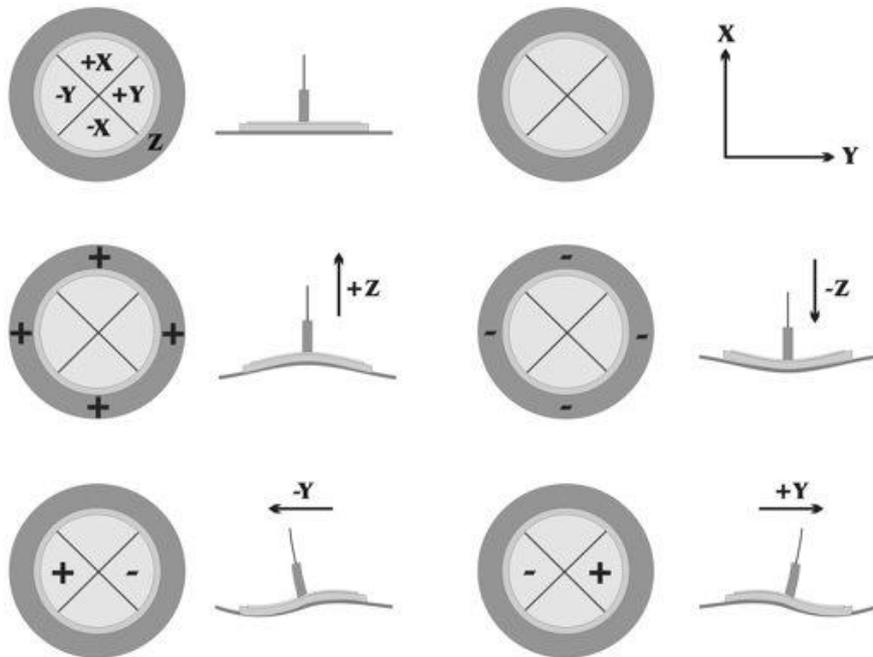


Abbildung 5: Funktions-Prinzip der Steuerung der Messspitze mittels modifiziertem Piezoschallgeber
 Quelle: Facharbeit, „Bau eines Rastertunnelmikroskops“ von Jonas Dittmann 2006, S.11

In Abbildung 5 wird dargestellt, wie die Steuerung der Messspitze funktioniert. Durch das Anlegen von Spannung an den jeweiligen Teil des Piezoschallgebers, verformt sich dieser entsprechend der oben dargestellten Grafik. Zwar wäre eine Auslenkung der Messspitze mit einseitigem Anlegen einer Spannung möglich, jedoch verstärkt sich die Auslenkung bei gegenläufig angelegter Spannung (siehe Abbildung 5, unten).

3.1. Schwingungsdämpfung der Scaneinheit

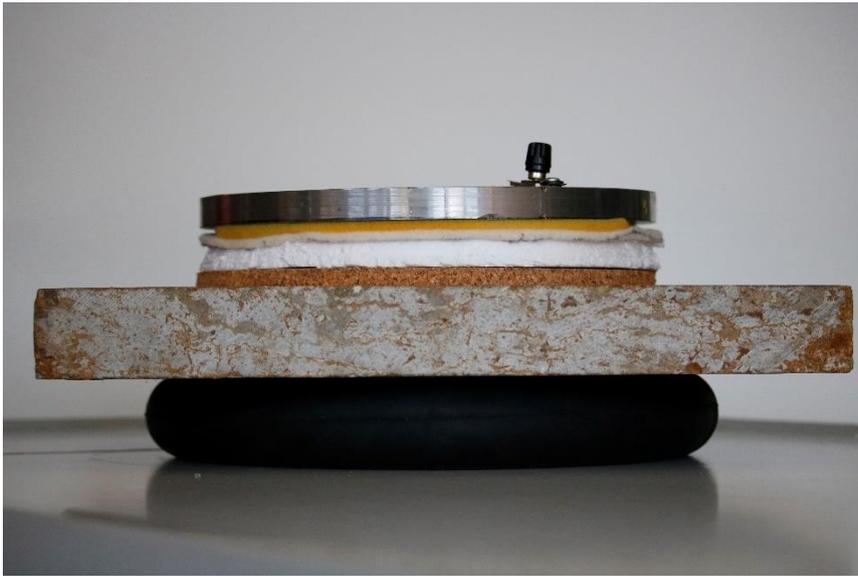


Abbildung 6: Schwingungsdämpfeinheit: Einsatz verschiedener Materialien zur Dämpfung von Bodenschwingungen.

Wie auf der Abbildung 6 zu sehen, wird die Scaneinheit auf mehreren, schwingungsdämpfenden Untergründen gelagert. Zur Dämpfung niederfrequenter Schwingungen wird ein schwach aufgeblasener Gummischlauch eingesetzt, auf welchen eine schwere Steinplatte gelegt wird. Durch das hohe Gewicht der Steinplatte, nutzt man zusätzlich die Trägheit der Platte zur Dämpfung. Für höherer Frequenzen wird zwischen Grundplatte (Metallplatte) der Scaneinheit und Steinplatte, eine Schicht Kork (Dicke 10mm), eine Schicht Styropor (Dicke 15mm), eine Schicht Filz (Dicke 4mm), sowie eine Lage Schaumstoff (Dicke 9mm) gelegt. Hierzu muss angemerkt werden, dass für die Dämpfungswirkung keine Messungen durchgeführt wurden, sondern lediglich eine intuitive Maßnahme ist, die sich, nach ansonsten dazu verwendeten Materialien richtete (vgl. (MISUMI Europa GmbH, 2016)).

3.2. Funktionsweise des Vorverstärkers

Da der Tunnelstrom, welcher fließt, „in der Größenordnung von 1 nA liegt“ (Universität Ulm, 2016) und daher extrem anfällig für äußere Störeinflüsse (elektromagnetische Störfelder) ist, muss dieser möglichst nahe an der Messspitze verstärkt werden. Hierzu wurde ein Operationsverstärker des Typs TLC2201 eingesetzt, welcher die Funktion hat, den Tunnelstrom zu verstärken und in eine Spannung umzuwandeln, da diese leichter auf verschiedene Sensoren, hier der Regelkreis, sowie der Arduino Due zu verteilen ist. Im weiteren Verlauf dieser Arbeit wird dieses Spannung als „Tunnelstromäquivalent bzw. U_T “ bezeichnet.

Der TLC2201 (Datenblatt siehe (Texas Instruments Incorporated, 2016)) wurde gewählt, da für diesen Anwendungszweck ein sehr rauscharmer, hochpräziser Rail-to-Rail-in/out Operationsverstärker (Op), notwendig ist. Die Rail-to-Rail Funktion wird benötigt, da die an der Probe angelegte Spannung zwischen/gleich +/- 5V liegt, was der Versorgungsspannung des Operationsverstärkers entspricht. Diese Eigenschaft definiert eine verzerrungsfreie Verarbeitung der Signale, bis hin zur Versorgungsspannung, was bei Operationsverstärkern ohne diese Funktion nicht der Fall ist (vgl. (Texas Instruments Incorporated, 2016)).

Als Grundlage der Berechnungen wurde das Kapitel 16, des Buches „Arbeitshilfen und Formeln für das technische Studium 4“ genutzt (vgl. (Böge, et al., 2013)).

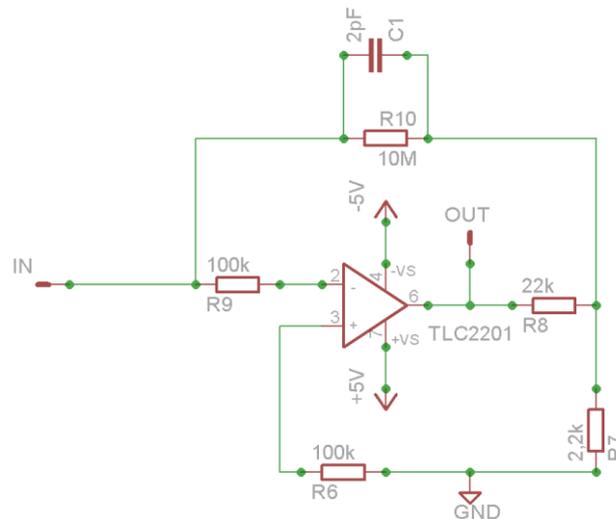


Abbildung 7: Schaltplan der Vorverstärkerschaltung

Der Stromspannungswandlungsfaktor der in Abbildung 7 dargestellten Schaltung, kann mittels R_{10} bestimmt werden. Für die Ausgangsspannung gilt:

$$U_T = I_{Tunnelstrom} * R_{10} \quad (1)$$

Da die Ausgangsspannung zwischen 0V und 5V liegt, folgt für den Tunnelstrom bei einer Ausgangsspannung von 2,5V:

$$\frac{U_T}{R_{10}} = I_{Tunnelstrom} = \frac{2,5V}{10M\Omega} = 2,5 * 10^{-7} A \quad (1.1)$$

Da jedoch der Tunnelstrom nicht mehr als 1nA betragen sollte, liegt dieser Wert beim 250-fachen des Solls. Dies deckt sich auch mit der Erfahrung, dass die Vorspannung extrem niedrig (nahe am Anschlag des Potentiometers) eingestellt werden muss.

3.3. Funktionsweise des Regelkreises (Z-Regelung)

Wie oben bereits beschrieben, ändert sich während der Rasterung der Probe ständig der Abstand zwischen Spitze und Probe. Da dies jedoch nicht gewollt ist, muss die Höhe der Spitze kontinuierlich nachgeregelt werden. Hier wurde dies mittels zweier Operationsverstärker in Kombination von Widerständen und Kondensatoren realisiert, welche dafür sorgen, dass der Tunnelstrom, der von Probe zu Spitze beziehungsweise umgekehrt fließt, konstant gehalten wird.

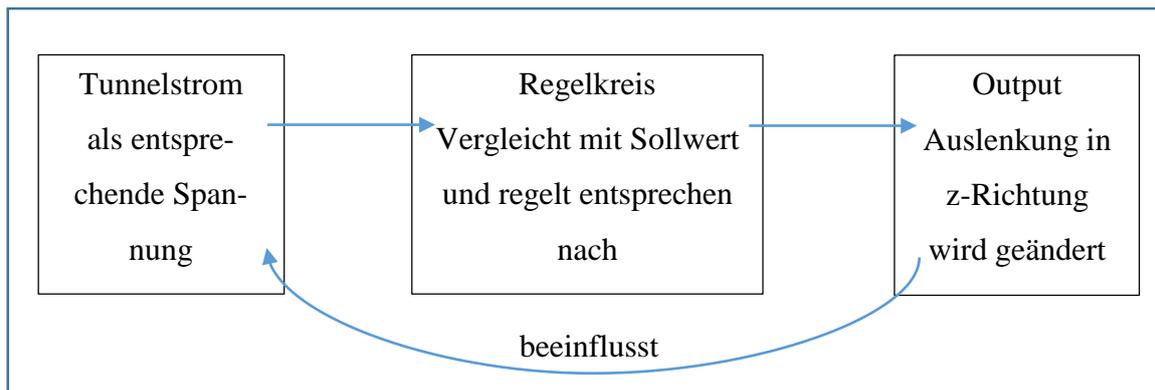


Abbildung 8: Regelkette schematisch

Der erste Operationsverstärker IC1A agiert dabei als aktiver Tiefpassfilter (TP) erster Ordnung, dieser ist notwendig, um auftretendes Rauschen während der Messung zu verhindern.

Um dies zu realisieren, wurde als Eingangswiderstand $R_3 = 10k\Omega$ am invertierten Eingang und als Rückkopplungswiderstand $R_1 = 10k\Omega$, zwischen invertiertem Eingang und Ausgang, gewählt. Parallel zum Rückkopplungswiderstand ist ein Kondensator $C_1 = 100pF$ geschaltet.

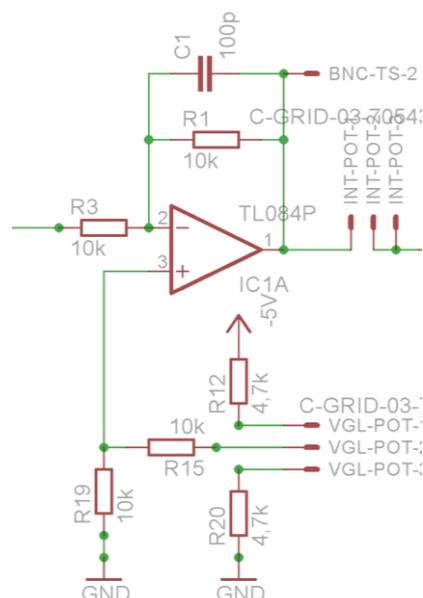


Abbildung 9: Aktiver Tiefpassfilter - Erstes Element der Regelkette (Ausschnitt aus Schaltplan im Anhang „Schaltplan neue Version.png“)

Als Grundlage der Berechnungen wurde das Kapitel 16, des Buches „Arbeitshilfen und Formeln für das technische Studium 4“ genutzt (vgl. (Böge, et al., 2013)).

Daraus ergeben sich für den aktiven Tiefpassfilter folgende Werte:

Verstärkungsfaktor V_{IC1A} :
$$V_{IC1A} = -\frac{R_2}{R_1} = -1 \quad (2)$$

Zeitkonstante τ_{TP} des Tiefpassfilters berechnet sich durch:

$$\tau_{TP} = R_2 * C_1 = 10k\Omega * 100pF = 1\mu s \quad (3)$$

Damit resultiert die **Grenzfrequenz f_c** mit $\tau_{TP} = \frac{1}{2\pi f_c}$ (4)

$$\Rightarrow f_c = \frac{1}{2\pi\tau_{TP}} \approx 159kHz \quad (4.1)$$

Der Tiefpassfilter schwächt also Abweichungen von der Sollspannung, welche mit einer höheren Frequenz als 159kHz auftreten, dabei steigt die Abschwächung mit zunehmender Frequenz an.

Das am nicht invertierten Eingang angeschlossene Potentiometer $R_{VGL-Pot\ max} = 20k\Omega$, mit angelegter Spannung von $U = -5V$, ist notwendig, um dem Tiefpassfilter eine Sollspannung bzw. Vergleichsspannung U_{Vgl} vorzugeben. Diese liegt zwischen 0V und -5V und entspricht dem Sollwert des Tunnelstromäquivalent.

An zweiter Stelle der Regelkette folgt ein zweiter Operationsverstärker IC1B, welcher als Integrierer geschaltet ist.

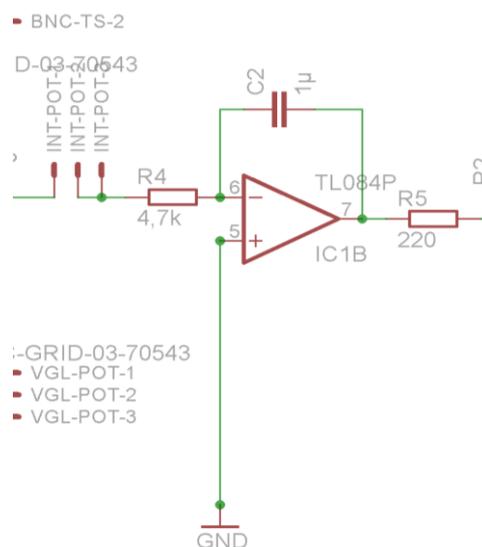


Abbildung 10: Integrierer - Zweites Element der Regelkette (Ausschnitt aus Schaltplan im Anhang „Schaltplan neue Version.png“)

Hierfür wurde ein Potentiometer mit $R_{INT-Pot\ max} = 100k\Omega$, sowie $R_4 = 4,7k$, also $R_{Ges} = 104,7k\Omega$ und ein Kondensator $C_2 = 1\mu F$ eingesetzt. Daraus resultiert bei Annahme fehlerfreier Widerstände und einem Potentiometer, welches auf die Hälfte des maximalen Wertes eingestellt ist, eine Zeitkonstante $\tau_{Int} = 54,7k\Omega * 1\mu F = 5,8ms$, welche den Integrator charakterisiert. Sie gibt an, wie lange es dauert, bis der Integrator die volle Ausgangsspannung erreicht hat.

Für die Ausgangsspannung U_A des Integrators gilt dann:

$$U_A = \frac{1}{\tau_{Int}} * \int U_E dt \quad (5)$$

Zu beachten ist, dass mit dem verbauten Potentiometer die Zeitkonstante des Integrierers beeinflusst wird, also die Nachregelungsgeschwindigkeit eingestellt werden kann.

Wird die Zeitkonstante des Integrators zu groß gewählt, so kann es sein, dass die Nachregelung zu träge ist und dadurch die Messspitze gegen die Probe prallt. Andererseits führt eine zu schnelle Nachregelung dazu, dass die Messspitze zu schwingen beginnt.

Zusammengefasst, verhindert das erste Glied der Regelkette ein Rauschen in der Messung, indem zu kleine Änderungen herausgefiltert werden. Das zweite Glied sorgt, bei richtiger Einstellung, für eine korrekte Nachregelung (nicht zu träge und nicht zu schnell). Da jedoch verschiedene Proben unterschiedlich beschaffen sind, ist es sinnvoll eine variable Nachregelungsgeschwindigkeit zu haben. So sollte bei zu messenden unebenen Oberflächen (zum Beispiel die Cluster eines DVD-Master), die Nachregelung besser etwas Träger, dagegen bei sehr feinen Strukturen (Goldfolie) eine empfindlichere Nachregelung gewählt werden.

3.4. Anpassung des Tunnelstromsignals und der X/Y-Steuerung an den Arduino Due

Der operative Bereich der analogen Messeingänge des Arduino Due liegt zwischen 0V und 3,3 V. Die ausgegebene Spannung der DAC-Output Pins liegt zwischen 0,55V und 2,75V.

Damit der Arduino Due verwendet werden kann, um sowohl den Tunnelstrom zu messen als auch die Messspitze zu steuern, muss also die jeweilige Spannung so angepasst werden, dass diese eine maximale Auflösung gewährleistet. Das heißt die Steuerspannung, welche bei 0,55V beginnt und bei 2,75V endet, muss so verschoben werden, dass bei 0,55V vom Arduino Due ausgegebener Spannung, -12V am Piezo, beziehungsweise am invertierten Eingang der jeweiligen Raumrichtung +12V, sowie bei 2,75V, +12V beziehungsweise, am invertierten Eingang -12V anliegen. Dies wurde mittels dreier Operationsverstärker gelöst.

Als Grundlage der Berechnungen wurde das Kapitel 16, des Buches „Arbeitshilfen und Formeln für das technische Studium 4“ genutzt (vgl. (Böge, et al., 2013)).

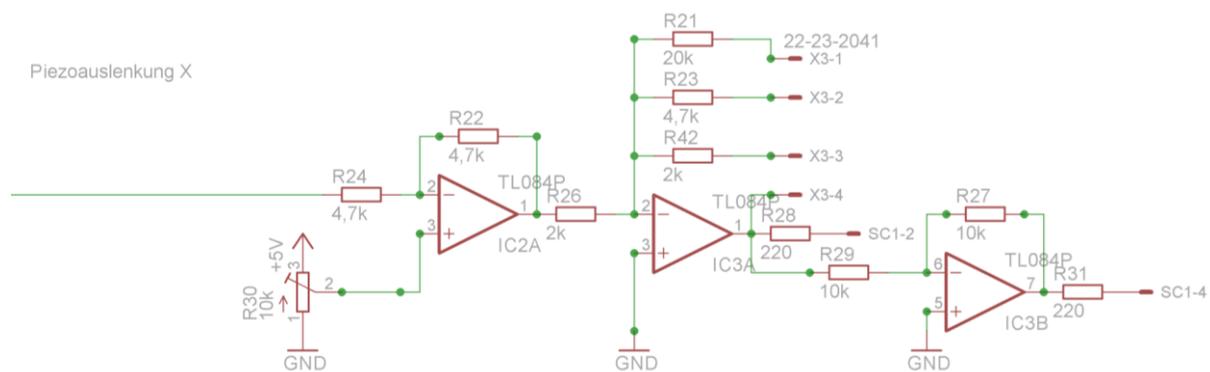


Abbildung 11: Piezoauslenkung Schaltplan in X-Richtung (bzw. Y-Richtung)

In Abbildung 11 ist der Schaltplan der Steuerung in X- beziehungsweise Y-Richtung dargestellt. IC2A agiert dabei als einfacher Spannungsverstärker mit Verstärkungsfaktor V_{IC2A} .

$$V_{IC2A} = -\frac{R_{22}}{R_{24}} = -1 \quad (6)$$

Durch die, am nicht invertierten Eingang angelegte Spannung, wird die Eingangsspannung, je nach Einstellung des Potentiometers R_{30} um bis zu 5V in negativer Richtung verschoben.

Diese muss jedoch nur um $U_1 = 1,1V$ in negativer Richtung verschoben werden, sodass bei 1,1V keine Spannung am Piezo anliegt (Definition des Nullpunkts).

Der einzustellende Wert lässt sich folgendermaßen berechnen:

$$R_{Pot} = R_1 + R_2 \quad (7)$$

Dabei beschreibt R_1 den Widerstand zwischen Pin 1 und 2, sowie R_2 den Widerstand zwischen Pin 2 und 3.

Da sowohl der Gesamtwiderstand als auch die Spannung bekannt ist, ergibt sich:

$$I = \frac{U_1}{R_{ges}} = \frac{5V}{10k\Omega} = 0,5mA \quad (8)$$

Da der Stromfluss in allen Bauteilen gleich groß ist (Reihenschaltung) folgt:

$$U_1 = I * R_1 \quad (9)$$

Daraus resultiert:

$$R_1 = \frac{U_1}{I} = \frac{1,1V}{0,5mA} = 2,2k\Omega \quad (9.1)$$

Somit fällt an $R_2 = R_{Pot} - R_1 = 7,8k\Omega$, 3,9V ab.

An zweiter Stelle, innerhalb der Schaltung, befindet sich der Operationsverstärker IC3A. Dieser ist als invertierender Verstärker geschaltet, dessen Verstärkung in drei Stufen wählbar ist, welches kleinere Ausschnitte in selber Auflösung zu untersuchen, ermöglicht.

Folgende Verstärkungsfaktoren stehen dabei zur Wahl:

$$V_{Zoom0} = -\frac{R_{42}}{R_{26}} = -10 \quad (10.1)$$

$$V_{Zoom1} = -\frac{R_{23}}{R_{26}} = -2,35 \quad (10.2)$$

$$V_{Zoom2} = -\frac{R_{21}}{R_{26}} = -0,5 \quad (10.3)$$

Der dritte Operationsverstärker ist ebenfalls als invertierender Verstärker mit dem Verstärkungsfaktor $V_{IC3B} = -1$ geschaltet und dient lediglich zur Investierung des Signals, um so die Spannung von Pin SC1-2 in invertierter Form an SC1-4 zu übergeben.

$$V_{IC3B} = -\frac{R_{27}}{R_{29}} = -1 \quad (11)$$

Die Schaltung in Y-Richtung ist analog zur Schaltung in X-Richtung.

Die tunnelstromäquivalente Spannung muss ebenfalls auf eine, durch den Arduino Due, erfassbare Größe gebracht werden. Dies wird ebenfalls mittels Operationsverstärker durchgeführt.

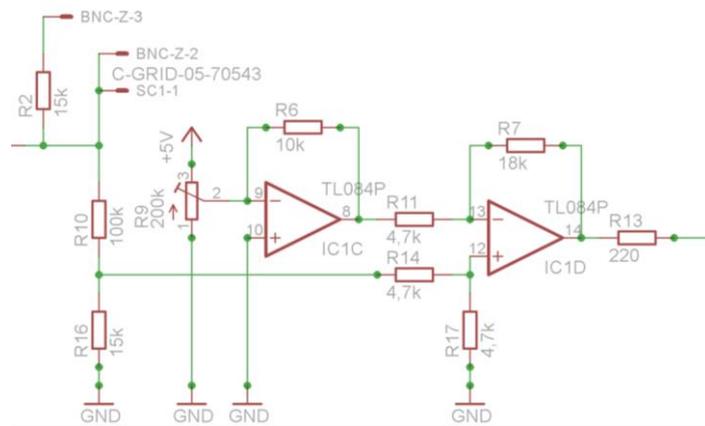


Abbildung 12: Ausschnitt aus Z-Regelung (Verschiebung und Verstärkung passend für Arduino Due)

In Abbildung 12 wird der Teil des Schaltplanes dargestellt, welcher dafür sorgt, dass die Nachregelspannung, welche am Piezo anliegt, in eine vom Arduino Due messbare Größenordnung gewandelt wird. Geht man von dem maximalen Bereich von -12V bis +12 Volt aus, so müssen diese auf den Messbereich von 0V bis 3V gewandelt werden, was mit zwei Operationsverstärkern realisiert wurde.

Der erste Operationsverstärker IC1C wurde als invertierender Verstärker geschaltet, sodass mittels R_9 eine „Sollspannung“ festgelegt werden kann. Diese legt quasi den Nullpunkt der Messwerte fest, sodass die gemessenen Schwankungen des Tunnelstromäquivalents um diesen Wert schwanken.

Zur Berechnung wird davon ausgegangen, dass am Ausgang -1,5V (Beispielwert) anliegen sollen.

Der Ausgangsspannung berechnet sich wie folgt:

$$U_{Aus} = \frac{R_6}{R_9 * x} * U_{Ein} \quad (12)$$

Mit $U_{Ein} = 5V$, $R_6 = 10k\Omega$, $R_9 = 200k\Omega$ und $U_{Aus} = -1,5V$ ergibt sich für die Schleiferposition x des Potentiometers:

$$x = -\frac{R_6}{R_9 * U_{Aus}} * U_{Ein} \quad (12.1)$$

$$x = -\frac{10k\Omega}{200k\Omega * (-1,5V)} * 5V = 0,1\bar{6} \quad (12.2)$$

Damit resultiert ein Verstärkungsfaktor von:

$$V_{IC1C} = -\frac{R_6}{R_9 * x} = -\frac{10k\Omega}{200k\Omega * 0,16} \approx -0,3 \quad (12.3)$$

Der Operationsverstärker IC1D ist als Differenzverstärker geschaltet. Werden alle Widerstände gleich groß gewählt, so gibt der Operationsverstärker lediglich die Differenz der beiden Eingangssignale aus.

In diesem Fall berechnet sich die Ausgangsspannung wie folgt:

U_a = Ausgangsspannung

U_{+E} = Eingangsspannung am nichtinvertierten Eingang

U_{-E} = Eingangsspannung am invertierten Eingang = U_{Aus} (IC1C)

$$U_A = \frac{(R_{11} + R_7) * R_{17}}{R_{14} + R_{17}} * U_{+E} - \frac{R_7}{R_{11}} * U_{-E} \quad (13)$$

Daraus folgt:

$$U_A = \frac{(4,7k\Omega + 18k\Omega) * 4,7k\Omega}{(4,7k\Omega + 4,7k\Omega) * 4,7k\Omega} * U_{+E} - \frac{18k\Omega}{4,7k\Omega} * U_{-E} \quad (13.1)$$

$$U_A = \frac{106,69M\Omega}{44,18M\Omega} * U_{+E} - \frac{18k\Omega}{4,7k\Omega} * U_{-E} \quad (13.2)$$

Die Ausgangsspannung ist dann:

$$U_A = 2,41 * U_{+E} - 3,83 * U_{-E} \quad (13.3)$$

Wie zu erkennen wird bei bilden der Differenz, die Spannung am invertierten Eingang stärker gewichtet, als die des nicht invertierten Eingangs.

3.5. Der Schrittmotor

Ein ständiger Wechsel der Messspitzen gehört zum regulären Ablauf mehrerer Messungen und muss daher im Verlauf der praktischen Arbeit sehr oft getauscht werden. Die selbstgefertigten Messspitzen weisen jedoch niemals dieselbe Länge auf, wodurch es notwendig ist, diese von Hand grob anzunähern, was mittels zweier Grobgewindeschrauben geschieht. Zur Feinannäherung an die Probe wird eine Feingewindeschraube eingesetzt, was sich bei manuellem Drehen jedoch als sehr unpraktisch erwies, da die Erschütterung sowie der Druck, die von Hand ausgeübt werden, genügen, den Regelbereich des Piezos zu überschreiten, wodurch in sehr vielen Fällen ein Crash mit der Probe verursacht wird. Dies veranlasste dazu eine erschütterungsärmere Variante der Annäherung zu finden. Da die Grobannäherung (bis auf etwa 0,5mm Abstand zur Probe) kein Problem darstellt, wird dies weiterhin von Hand durchgeführt. Die Feinannäherung wird nun jedoch von einem Schrittmotor Typ KP4M2-203 (vgl. (Conrad Electronic SE , 2016)) übernommen, welcher mittels Schlauchkupplung und eingestecktem Sechskantschlüssel die Feingewindeschraube bewegt. Simultan wird der Tunnelstrom mittels Arduino Due gemessen. Erreicht der Strom einen gewissen Wert, so wird der Motor gestoppt. Näheres hierzu findet sich im Kapitel 6.

Das Ganze ist wie folgt aufgebaut:

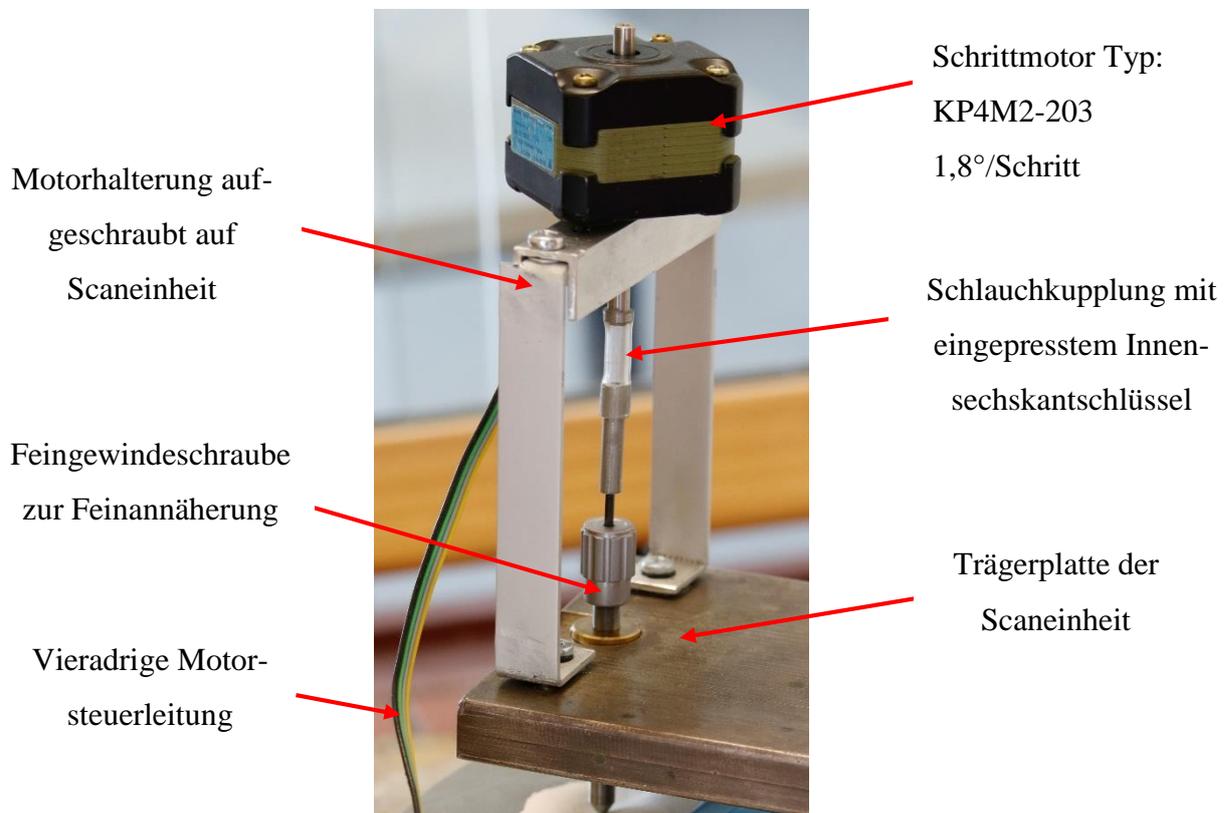


Abbildung 13: Schrittmotor mit Haltevorrichtung

Angesteuert wird der Schrittmotor dabei mit einem Schrittmotortreibermodul, welches vom Arduino Due entsprechende Steuersignale empfängt. Diese enthalten Informationen, wie Schrittweite (Regelbar von ganzen, 1/2, 1/4, 1/8 und 1/16 Schritten), Start und Stopp-Signale sowie ein Signal zum Deaktivieren des Motors (vgl. (Pololu Corporation, 2001-2016)). Das Deaktivierungssignal wird hier explizit genannt, da im Motor bei einem Stoppsignal elektromagnetische Gegenfelder angelegt werden, wodurch der Rotor blockiert wird. Das kann einerseits zur Überhitzung des Motors, andererseits zu störender, elektromagnetischer Strahlung führen. Die untenstehende Grafik zeigt die Verdrahtung des verwendeten Schrittmotor-Treibermoduls, welches ein identischer Nachbau des Polulu A4988 ist.

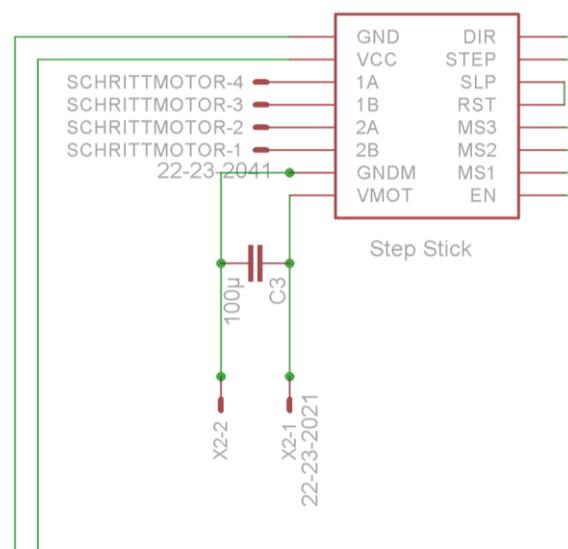


Abbildung 14: Schrittmotor- Treibermodul Baugleich zu Polulu A4988 mit weiterführenden Anschlüssen („Ausschnitt aus Schaltplan neue Version.png“)

Das Modul wird direkt auf die Platine aufgesteckt, was den Vorteil bietet, dass einige Leitungen durch Leiterbahnen ersetzt werden können. So werden beispielsweise sämtliche Pins der rechten Seite, sowie die Pins „GND“ und „VCC“, direkt via Leiterbahnen mit dem Arduino Due verbunden.

An „VCC“ wird eine Spannung von 3,3V angelegt. Alle Pins der rechten Seite des Schrittmotor-Treibermoduls werden mit digitalen Pins des Arduino Due verbunden. Die Pins „1A/B“, sowie „2A/B“ werden, mittels vieradriger Leitung mit angelötetem Stecker direkt mit dem Schrittmotor verbunden und versorgen diesen mit der elektrischen Energie. Die beiden Pins „GNDM“ und „VMOT“ führen zu den zwei Pins x2-1&2, an welche eine 12V Gleichspannungsquelle angeschlossen wird. Die Spannungsversorgung des Motors wird extern und nicht von der Board-Spannungsquelle ausgeführt, um mögliche Spannungsschwankungen innerhalb der Regelkette des STM zu vermeiden.

3.6. Platine

Wie bereits beschrieben, wurde die Platine aufgrund von einigen nicht auffindbaren Defekten neu hergestellt. Die neue Platine wurde jedoch mit einem neuen Layout versehen, sowie daran einige technische Abänderungen vorgenommen. „Ein Punkt für Punkt“ - Vergleich von altem und neuen Layout ist jedoch nicht sinnvoll, da die Platine vollständig dahingehend verändert wurde, dass jegliche Lötunkte, die Durchkontaktierungen ausgenommen, auf der Unterseite der Platine platziert sind. Daher sind die Leiterbahnen vollständig verschieden zur Vorgängerversion. Beide Versionen der Platine, sowie des Schaltplans, können im Anhang eingesehen werden („Platine bzw. Schaltplan neue Version.png“ und „Platine bzw. Schaltplan alte Version.png“). Anhand der Schaltpläne lassen sich jedoch die Veränderungen und Neuerungen sehr gut erläutern.

Im Bereich der Tunnelstromanalyse und Z-Nachregelung der Messspitze, wurde ein zusätzlicher Operationsverstärker IC2C (siehe Abbildung 16, Markierung) vorgeschaltet.

Tunnelstrom-Analyse -> Z-Auslenkung

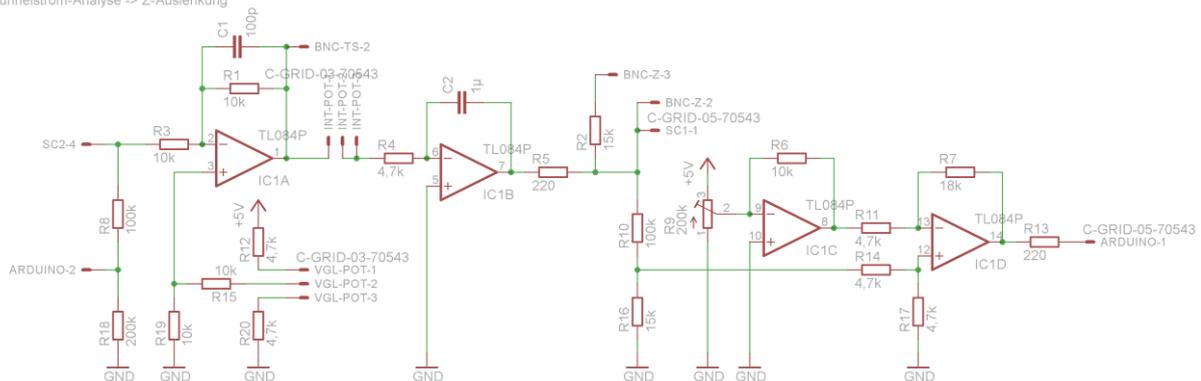


Abbildung 15: Schaltplanausschnitt - Tunnelstromanalyse und Z-Auslenkung der Messspitze (alte Version)

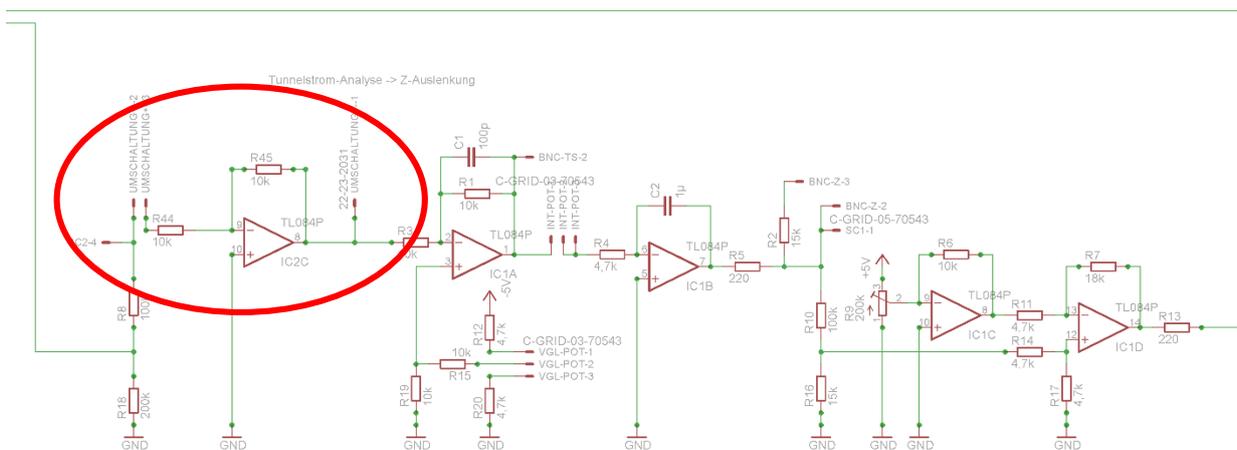


Abbildung 16: Schaltplanausschnitt - Tunnelstromanalyse und Z-Auslenkung der Messspitze (neue Version)

Dieser ist als Invertierer geschaltet und kann mittels Verbindung der Pins „Umschaltung +- 2“ und „Umschaltung +- 3“ zugeschaltet, beziehungsweise durch Verbinden der Pins „Umschaltung +- 2“ und „Umschaltung +- 1“ überbrückt, werden.

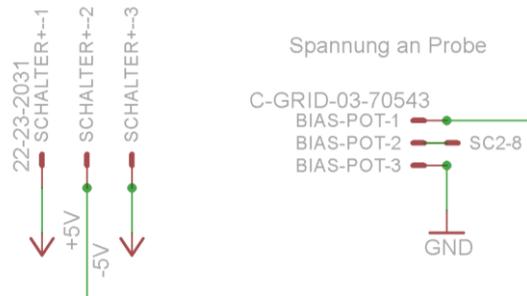


Abbildung 17: Spannung an Probe (neue Version)

Wird bei Zuschalten des Invertierers in Abbildung 16, gleichzeitig die Spannung an der Probe invertiert (Abbildung 17), indem die Pins „Schalter +- 1“ und „Schalter +-2“ verbunden werden, so ist es möglich, eine Messung mit positiver Spannung an der Probe durchzuführen. Bei Messungen mit negativer Spannung an der Probe, das bedeutet Pin „Schalter +-2“ ist verbunden mit Pin „Schalter +-3“, muss jedoch der Invertierer überbrückt werden. Diese Option wurde neu hinzugefügt.

Das in dieser Schaltung verbaute Potentiometer, agiert als Spannungsteiler, wodurch an die Probe eine Spannung von 0V bis 5V bzw. -5V bis 0V angelegt werden kann.

Eine weitere Neuerung ist, dass der genutzte Arduino Due nun nichtmehr mittels Kabelsteckverbindungen mit der Hauptplatine verbunden, sondern nun direkt auf diese aufgesteckt wird. Dies bietet einerseits den Vorteil, dass kürzere und damit störungsfreiere Signalübertragungswege gewährleistet sind, andererseits wird so, das vorher herrschende Kabelchaos auf ein Minimum reduziert und eine Fehlverdrahtung unterbunden.

Außerdem wurde, wie bereits im Kapitel 3.5 „Der Schrittmotor“ erwähnt, ein Steckplatz für das Schrittmotor-Treibermodul geschaffen, welches ebenfalls direkt in die restliche Elektronik eingebunden, also mit dem Arduino Due via Leiterbahnen verbunden, ist.

Zusätzlich findet sich eine Neuerung in der X/Y-Steuerung, wobei die Zoomfunktion um eine weitere Stufe erweitert wurde.

4. Spitzenherstellung

Die Messspitze des STM, ist der wichtigste Bestandteil des Rastertunnelmikroskops, da eine Messung mit schlechter Spitze unweigerlich zu schlechten beziehungsweise völlig unbrauchbaren Ergebnissen führt. Die „Spitzheit“, idealerweise einatomig am Spitzenende, und die Form der Spitze, je konischer desto besser, sind die wichtigsten Eigenschaften. Für die Herstellung solcher Spitzen eignen sich im Speziellen zwei Materialien sehr gut. Dies ist zum Einen, Platin/Iridium als Draht, welcher aber wegen seinem hohen Preis (ca.411€/2m) uninteressant ist, trotz, dass dieser den Vorteil hat, dass die Spitzen relativ einfach durch „auseinanderreißen“ des Drahtes hergestellt werden können, zum Anderen Wolframdraht. Dieser ist mit etwa 428€/200m weitaus günstiger (Preise für Draht mit einem Durchmesser von 0,25mm (Goodfellow GmbH, 2016)). Die nachstehende Grafik der Universität Hannover zeigt ein Vergleich zweier Spitzen, mit *links* geätzter Wolfram-Spitze und *rechts* gerissener Platin-Iridium-Spitze. Ein Elektrolyseätzverfahren ist ebenso mit Platin/Iridium Draht möglich.

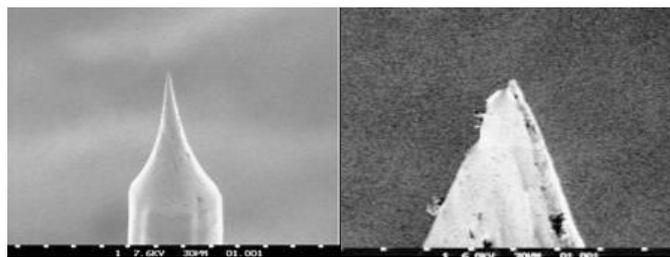


Abbildung 18: (links)Wolframspitze im Rasterelektronenmikroskop (geätzt)
(rechts) Platin/Iridiumspitze im Rasterelektronenmikroskop (gerissen)
Quelle: <http://www.iqo.uni-hannover.de/367.html>

Nachteilig bei Wolframspitzen ist jedoch, dass diese direkt vor dem Messvorgang hergestellt werden sollten, da diese relativ schnell, mit dem in der Luft vorhandenen Sauerstoff oxidieren. Das bedeutet eine Herstellung auf Vorrat ist nicht sinnvoll. Ein weiteres häufig auftretendes Problem, sind Verunreinigungen, sowie Wolframoxidreste auf der Spitze, welche die Qualität der Spitze beeinträchtigen können. So sollte die Spitze nach dem Ätzvorgang durch gereinigtes Wasser gezogen werden.

„Sowohl die rein mechanische als auch die elektrochemische Variante der Spitzenherstellung kann durch eine „in vivo“ - Nachbehandlung verbessert werde: Dadurch, dass man die Spitze einer elektrischen Spannung aussetzt kommt es an der Spitze zu enorm großen elektrischen Feldern, die eventuell noch an der Spitze haftendes Material im wahrsten Sinn des Wortes wegblasen, und somit die Spitzen noch schärfer, und somit geeigneter für STM machen“ (Sonnleitner, 2003).

Da solch hohe elektrische Felder unter normalen Bedingungen jedoch nicht möglich sind, wird die Spitze ohne Nachbehandlung genutzt. Dies verschlechtert zwar vermutlich das Messergebnis, liefert aber dennoch sehr brauchbare Resultate, wie Abbildung 2 zeigt.

Ein reliabler Vergleich der Unterschiede einer nachbehandelten und nicht-nachbehandelten Spitze wäre mit dem vorhandenen Versuchsaufbau nahezu unmöglich und wurde daher auch nicht durchgeführt. Natürlich gibt es noch weitere verwendbare Stoffe wie Gold, die sich für die Nutzung als Messspitzen eignen, jedoch sind Wolfram und Platin/Iridium die Gängigsten.

Zur Herstellung der Spitzen wird eine Spitzenätzstation verwendet (vgl. (Reuß, 2011))

Diese bestand schon vor Beginn der Arbeit und wurde bis auf eine kleine Abänderung, bezüglich des leichteren Einspannens des zu ätzenden Drahtes, nicht verändert. Ursprungszustand war, dass der Draht mittels Anziehen einer Schraube mit einem Schraubenzieher fixiert werden musste. Diese Schraube wurde durch eine von Hand drehbare Schraube ersetzt, wodurch eine leichtere, genauere Positionierung des zu ätzenden Drahtes möglich ist.

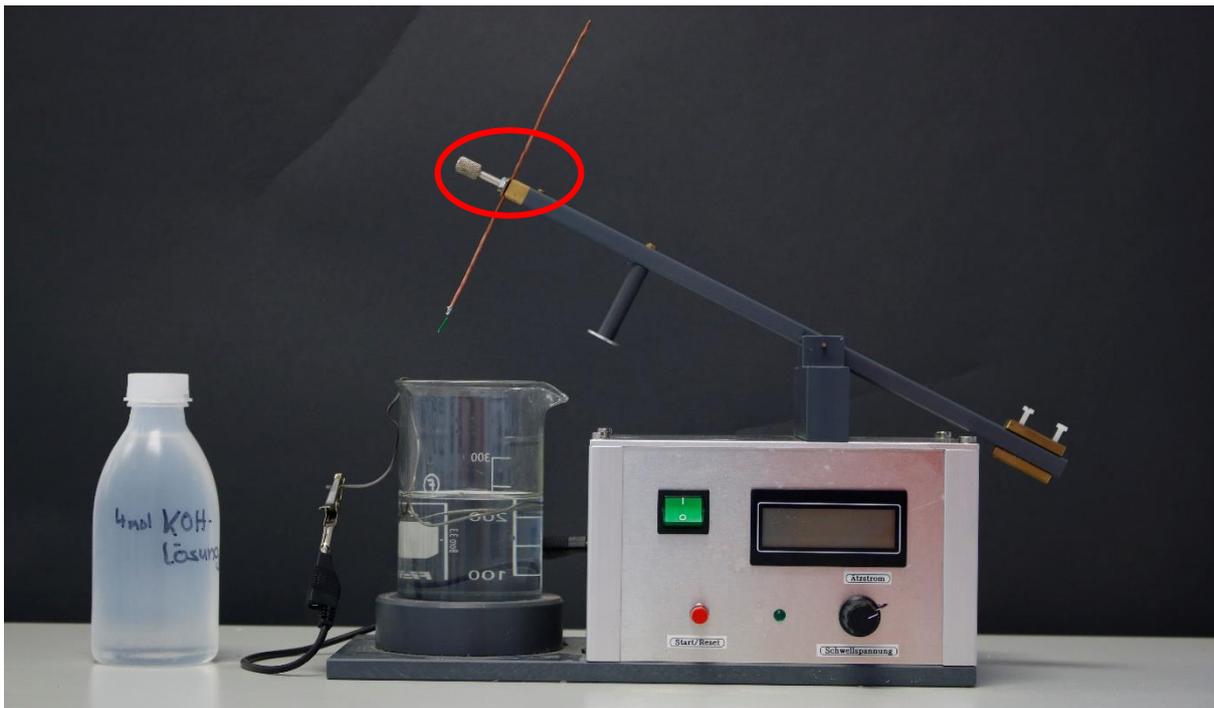


Abbildung 19: Gerätschaft zur Herstellung von STM-Spitzen

In die, im Bild markierte Einspannvorrichtung, wird der zu ätzende Draht, hier, ein an einen Kupferdraht ($l \approx 15\text{cm}$, $\varnothing = 2\text{mm}$) angelöteten Wolframdraht ($l \approx 1,5\text{cm}$, $\varnothing = 0,2\text{mm}$), eingespannt. Dieser fungiert später als Anode, da die Einspannvorrichtung gleichzeitig der Anodenanschluss der Ätzstation ist. In das Glasgefäß wird eine vier-molare KOH-Lauge bis etwa zur Hälfte eingefüllt. Dabei sollte bei der Herstellung der Lauge darauf geachtet werden, dass diese mit gereinigtem Wasser hergestellt wird und im Idealfall nicht längere Zeit offensteht, da

ansonsten, das in der Luft vorhandenen Kohlenstoffdioxid, die Lauge verunreinigen und sich diese in Pottaschen-Lösung umwandelt.



Anschließend wird die Kathode (Ring aus dickem Kupferdraht) in das Bad eingetaucht und mittels einer Krokodilklemme mit dem Kathodenanschluss der Ätzstation verbunden. Nun wird das Gerät eingeschaltet und mit dem Startknopf eine Gleichspannung von ca. 5V angelegt. Daraufhin wird der Kipparm abgesenkt. Durch einen Elektromagneten wird dieser in horizontaler Position gehalten. Nun sollte das rechte Gegengewicht so eingestellt werden, dass der Elektromagnet den Arm gerade noch hält. Dies ist deshalb wichtig, dass bei fertigem Ätzvorgang, die Spitze möglichst schnell aus der Lauge gezogen wird. Nun wird der Kipparm in waagerechter Stellung gehalten und der Kupferdraht mit angelötetem Wolframdraht (Anode) soweit in der Einspannvorrichtung abgelassen, dass sich lediglich 2-4mm des Wolframdrahtes oberhalb der KOH-Lauge befinden. Nun kann der Kipparm vorsichtig losgelassen werden.

Durch die angelegte Spannung zwischen Anode und Kathode wandern die OH^- -Ionen zur Anode und oxidieren das Wolfram (vgl. (Dittmann, 2006)). Durch die Schwerkraft und die Oberflächenspannung der Lauge fließt das oxidierte Wolfram nach unten ab und es entsteht ein immer dünner werdendes Zwischenstück. Dies geschieht so lange, bis die am unteren Stück angreifende Schwerkraft genügt, die Verbindung abzureisen. In diesem Moment ändert sich der Widerstand der Schaltung schlagartig, was von der Elektronik erkannt wird, welche daraufhin den Elektromagneten deaktiviert, weshalb der Kipparm nach oben schnellt und den oberen Teil des geätzten Drahtes aus der Lauge zieht.

Die dadurch erhaltene Spitze kann anschließend mit einem Lichtmikroskop, recht gut auf ihre Qualität untersucht werden.

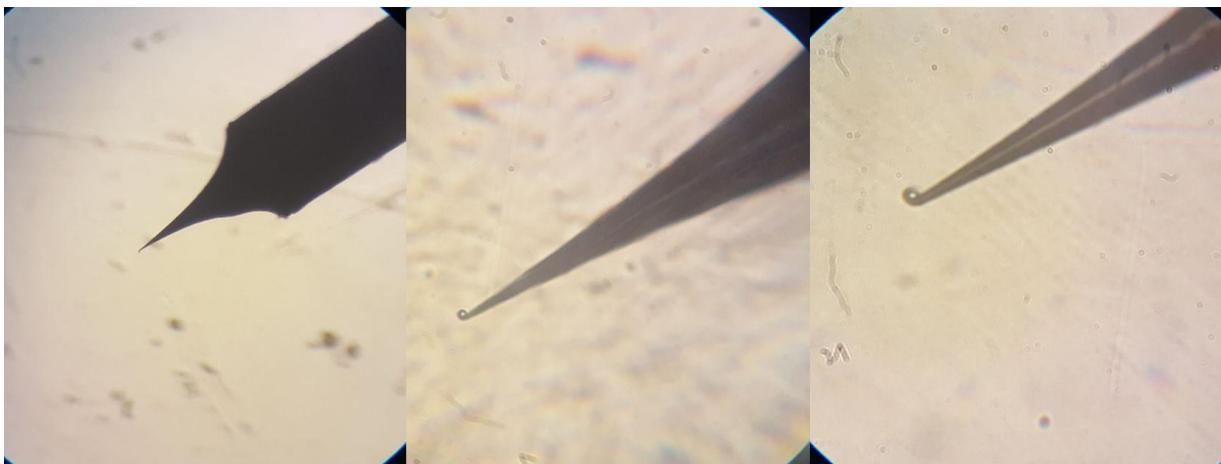


Abbildung 20: Spitze nach Stoß mit einer Probe unter dem Lichtmikroskop
- von links nach rechts: 100-fache Vergrößerung, 600-fache Vergrößerung, 1000-fache Vergrößerung

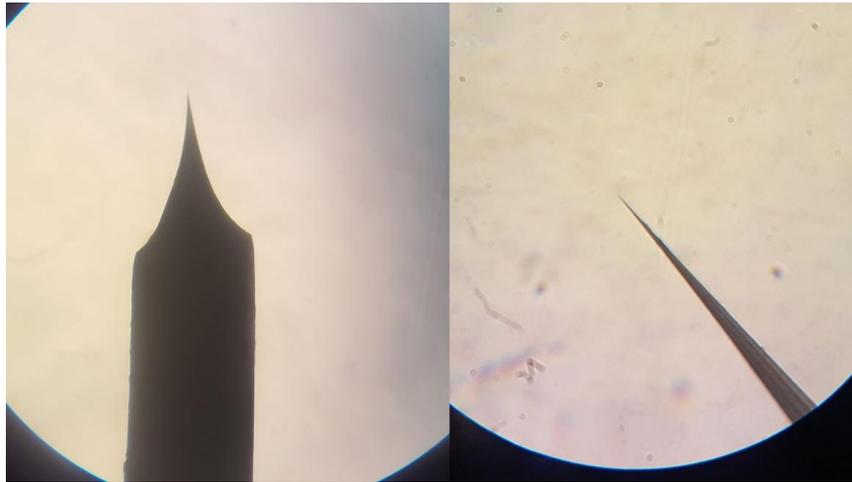


Abbildung 21: Frisch geätzte Spitze, links 100-fache Vergrößerung, rechts 600facher Vergrößerung

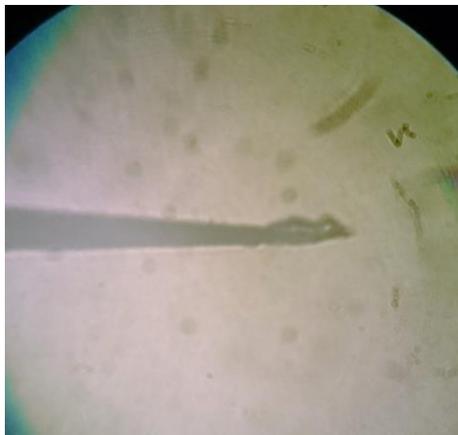


Abbildung 22: schlecht geätzte Spitze, nach zu frühem Auslösen, bzw. Abriss an falscher Stelle, bei 1000facher Vergrößerung

Die genaue Vorgehensweise zur Herstellung einer Spitze wird in Kapitel 7.2 näher erläutert.

5. Überprüfung der Platine

Die ursprüngliche Aufgabe der Arbeit war es, die einzelnen Elemente des STM auf korrekte Funktion zu überprüfen. Dies wurde mittels dem Sensor „Cassy 2“ und CassyLab 2, sowie zweier Labornetzgeräte (EA-PS 2016-100), einem Oszilloskop (Tektronix TDS2012B) und einem Frequenzgenerator (Metrix ORX 310), realisiert. Die Messungen wurden an der ursprünglichen Platine vollzogen. Zwar wurde die neue Version ebenfalls überprüft, dies wurde jedoch nur zu Einstellzwecken der verbauten Potentiometer vollzogen und nicht dokumentiert. Es wurde überprüft, ob das Verhalten des Regelkreises, sowie die Verschiebung und Verstärkung der Arduino Due-Signale zur Bewegung des Piezos, als auch das Signal des Tunnelstromäquivalents an den Arduino Due, dem Gewünschten entsprach.

Hierzu benötigt es einiger Daten zum Arduino Due (vgl. (Arduino, 2016));

- Die Ausgangsspannung der DAC -Pins, welche die Steuerung der X- sowie Y-Richtung übernehmen, liegt zwischen 0,55V und 2,75V, also deckt somit einen Bereich von 2,2V ab. Diese Spannung kann in 4095 Schritte unterteilt werden und ergibt somit eine Schrittweite von 0.5372mV. Das bedeutet letztendlich für das Rastertunnelmikroskop, dass die maximale Auflösung knapp 16 Megapixel beträgt. Also maximal 4096x4096 Punkte angefahren werden können.
- Die analogen Eingangspins besitzen einen Messbereich von 0V bis 3,3V

5.1. Überprüfen der X-Steuerung

Der vom Arduino Due ausgegebene Spannungsbereich von 0,55V bis 2,75V (Eingangsspannung U_e) muss so verschoben und verstärkt werden, dass am Ausgang der Platine für die Steuerung des Piezos ein Spannungsbereich von -12 V bis 12 V (Ausgangsspannung U_a) ausgegeben wird. Das bedeutet, es muss beim Auftragen der Eingangsspannung gegen die Ausgangsspannung ein linearer Zusammenhang sichtbar sein.

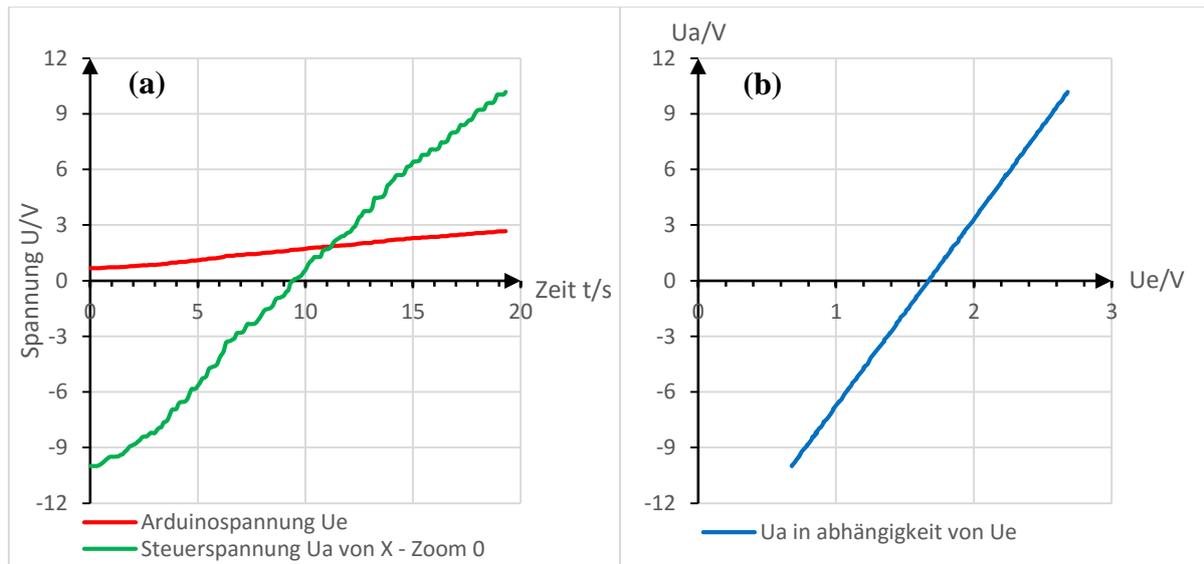


Diagramm 1: (a): Eingangsspannung(rot) und Ausgangsspannung (grün) in X-Richtung Zoomstufe 0
(b): Zusammenhang von Eingangs- zu Ausgangsspannung in X-Richtung Zoomstufe 0

In obigem Diagramm 1(a) wird die vom Arduino Due ausgegebene Eingangsspannung (grün) gemeinsam mit der resultierenden Ausgangsspannung (rot) dargestellt. Im rechten Graphen ist sehr gut der lineare Zusammenhang zwischen Eingangsspannung und Ausgangsspannung zu erkennen. Bei einem Wert von etwa 0,67V Eingangsspannung resultiert eine Ausgangsspannung von ca. -10,0V. Eine Eingangsspannung von ca. 2,67V resultiert in einer Spannung von ca. 10,1V. Die Abweichung zur Sollspannung 12V bzw. -12V stellt kein Problem dar, da sich dadurch lediglich der Bildausschnitt verkleinert. Wie erwartet, zeigt sich beim Auftragen der Eingangsspannung gegen die Ausgangsspannung in Diagramm 1(b) deutlich der lineare Zusammenhang beider Größen, mit einer Steigung von 10V Ausgangsspannung pro 1V Eingangsspannung.

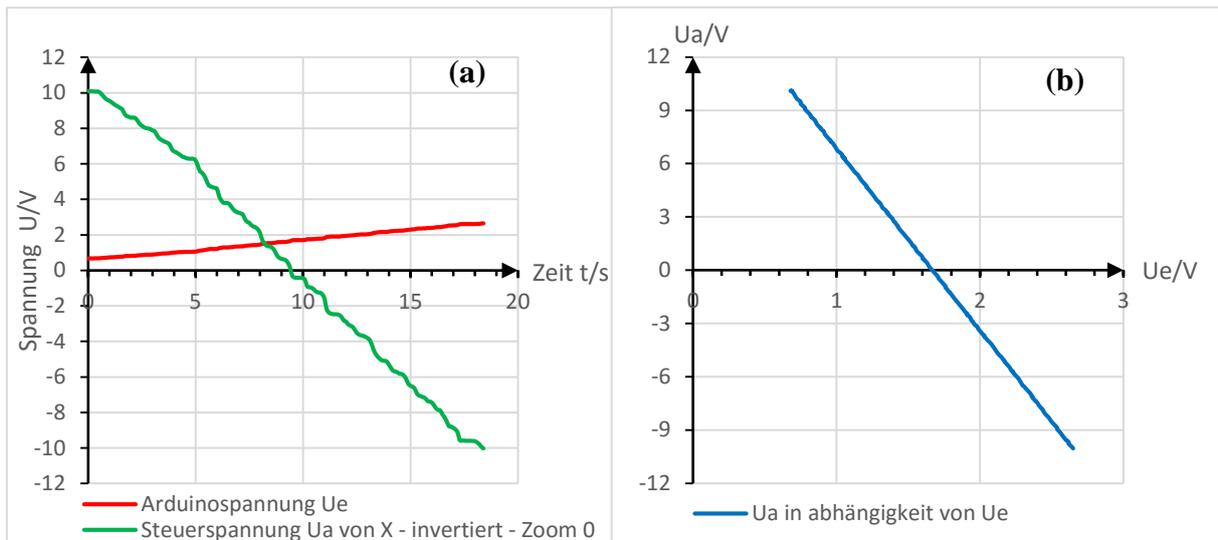


Diagramm 2: (a): Eingangsspannung (rot) und Ausgangsspannung invertiert (grün) in X-Richtung Zoomstufe 0
 (b): Zusammenhang von Eingangs- zu invertierter Ausgangsspannung in X-Richtung Zoomstufe 0

Diagramm 2 (a) zeigt die Eingangsspannung (grün) mit der invertierten Ausgangsspannung (rot). Es ist wie in Diagramm 1 zu erkennen, dass die Ausgangsspannung ebenfalls um je ca. 2V von der Sollspannung differiert, stellt aber ebenso kein Problem dar. Diagramm 2 (b) lässt auch deutlich den linearen Zusammenhang beider Größen, mit einer Steigung von -10V Ausgangsspannung pro 1V Eingangsspannung erkennen. Wie erwartet, zeigen beide (2(a) & 2(b)) Graphen, den invertierten Spannungsverlauf von Diagramm 1.

Anmerkung: Diagramm 1 und Diagramm 2 zeigen verschiedene Messungen.

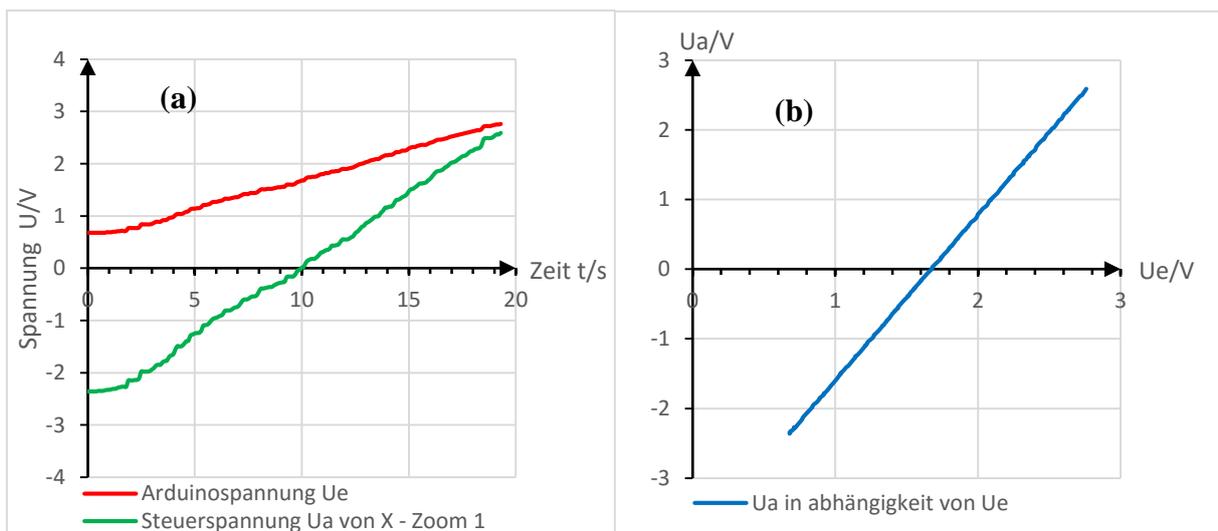


Diagramm 3: (a): Eingangsspannung (rot) und Ausgangsspannung (grün) in X-Richtung Zoomstufe 1
 (b): linearer Zusammenhang von Eingangs- zu Ausgangsspannung in X-Richtung Zoomstufe 1

In Diagramm 3(a) ist die Eingangsspannung (grün) und die resultierende Ausgangsspannung (rot) dargestellt. Hierbei beträgt die Ausgangsspannung jedoch nur ca. $-2,37\text{V}$ bis $2,82\text{V}$, dabei bleibt der Spannungsbereich der Eingangsspannung jedoch unverändert. Der kleinere Spannungsbereich folgt aus der Einstellung des Zooms. Wie zu erkennen, liegt keine Symmetrie zum Nullpunkt vor, sondern zeigt eine Verschiebung von etwa $0,45\text{V}$ in X-Richtung. Dies erweist sich zwar nicht als problematisch, da hierdurch lediglich der Bildausschnitt verschoben ist, was aber dennoch korrigiert wurde.

In Diagramm 3 (b) zeigt sich deutlich der lineare Zusammenhang beider Größen, mit einer Steigung von ca. $2,4\text{V}$ Ausgangsspannung pro 1V Eingangsspannung.

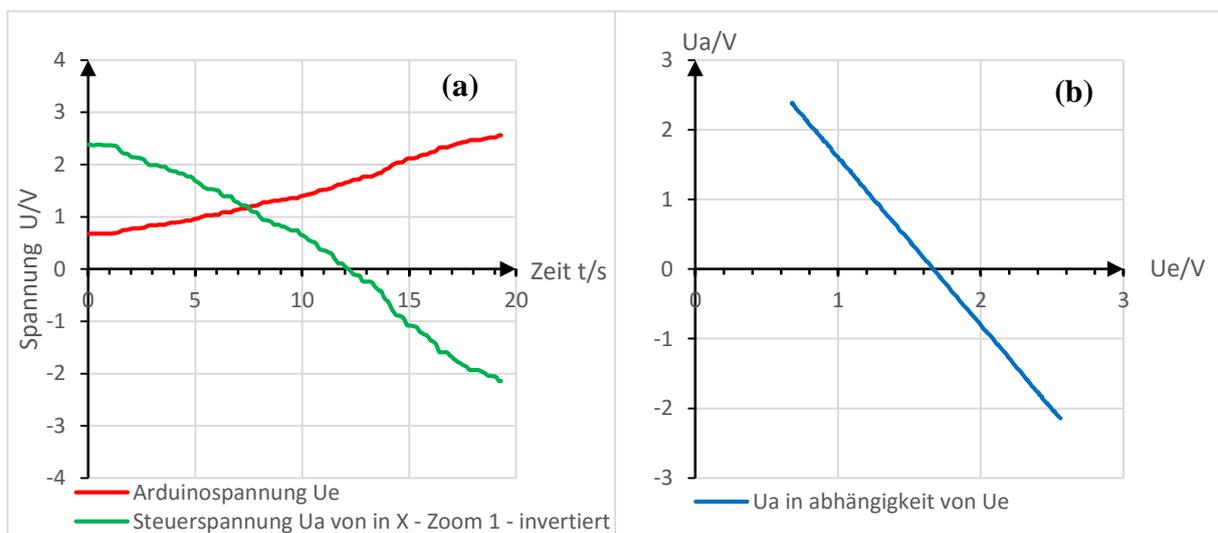


Diagramm 4: (a): Eingangsspannung (rot) und Ausgangsspannung invertiert (grün) in X-Richtung Zoomstufe 1
(b): Zusammenhang von Eingangs - zu invertierter Ausgangsspannung in X-Richtung Zoomstufe 1

Diagramm 4 (a) zeigt den invertierten Ausgang, das bedeutet, dass die Ausgangsspannung mit steigender Eingangsspannung absinkt. Das erreichte Minimum beziehungsweise Maximum der Ausgangsspannung, gleicht der Invertierten von Diagramm 3. In Diagramm 4 (b) zeigt sich deutlich der lineare Zusammenhang beider Größen, mit einer Steigung von ca. $-2,4\text{V}$ Ausgangsspannung pro 1V Eingangsspannung.

Anmerkung: Diagramm 3 und Diagramm 4 zeigen verschiedene Messungen.

5.2. Überprüfen der Y-Steuerung

Auf eine Erläuterung der Graphen wird in diesem Abschnitt verzichtet, da diese analog zu den Erklärungen von Kapitel 5.1 sind.

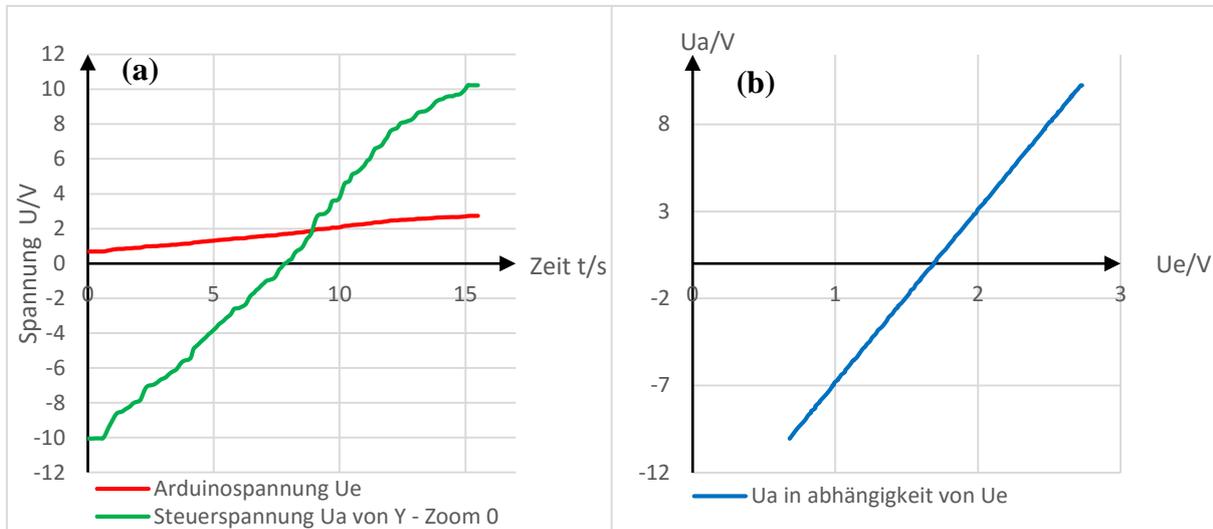


Diagramm 5: (a): Eingangsspannung (rot) und Ausgangsspannung (grün) in Y-Richtung Zoomstufe 0
 (b): Zusammenhang von Eingangs- zu Ausgangsspannung in Y-Richtung Zoomstufe 0

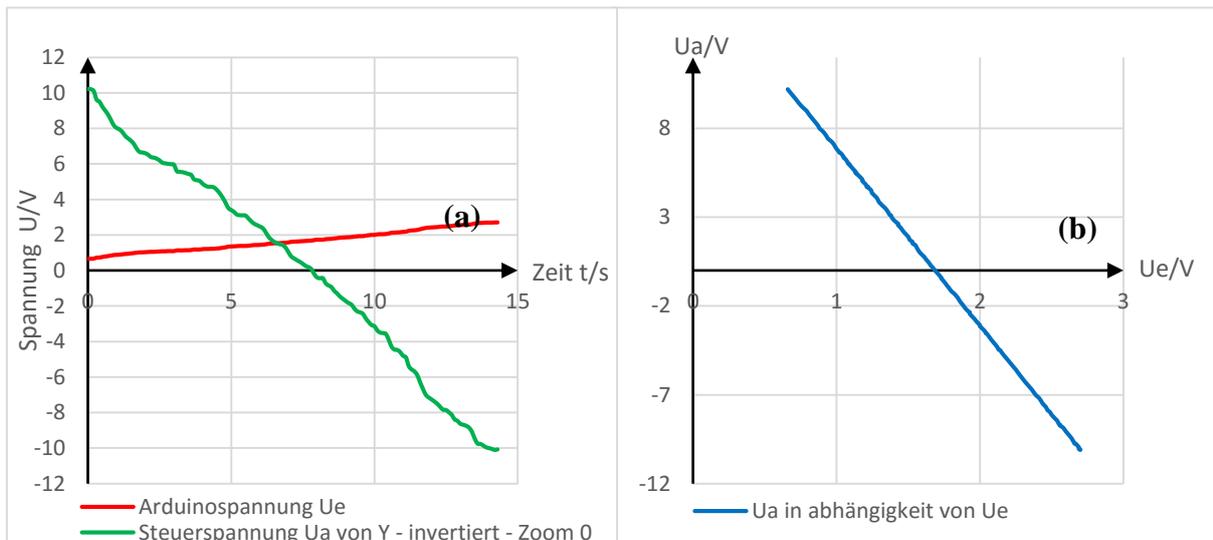


Diagramm 6: (a): Eingangsspannung (rot) und Ausgangsspannung invertiert (grün) in Y-Richtung Zoomstufe 0
 (b): Zusammenhang von Eingangs- zu invertierter Ausgangsspannung in Y-Richtung Zoomstufe 0

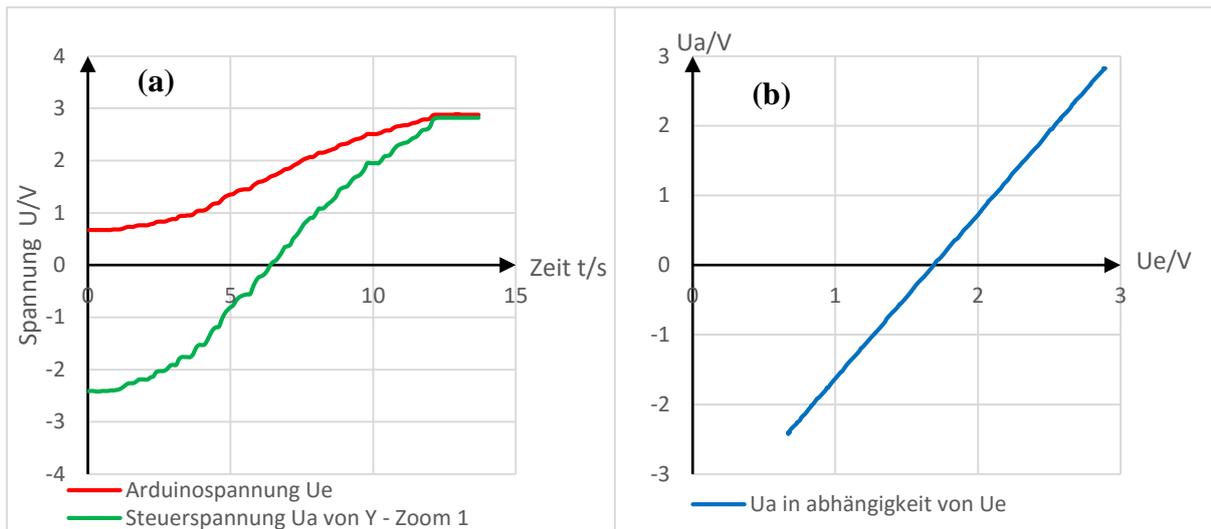


Diagramm 8: (a): Eingangsspannung (rot) und Ausgangsspannung (grün) in Y-Richtung Zoomstufe 1
 (b): Zusammenhang von Eingangs- zu Ausgangsspannung in Y-Richtung Zoomstufe 1

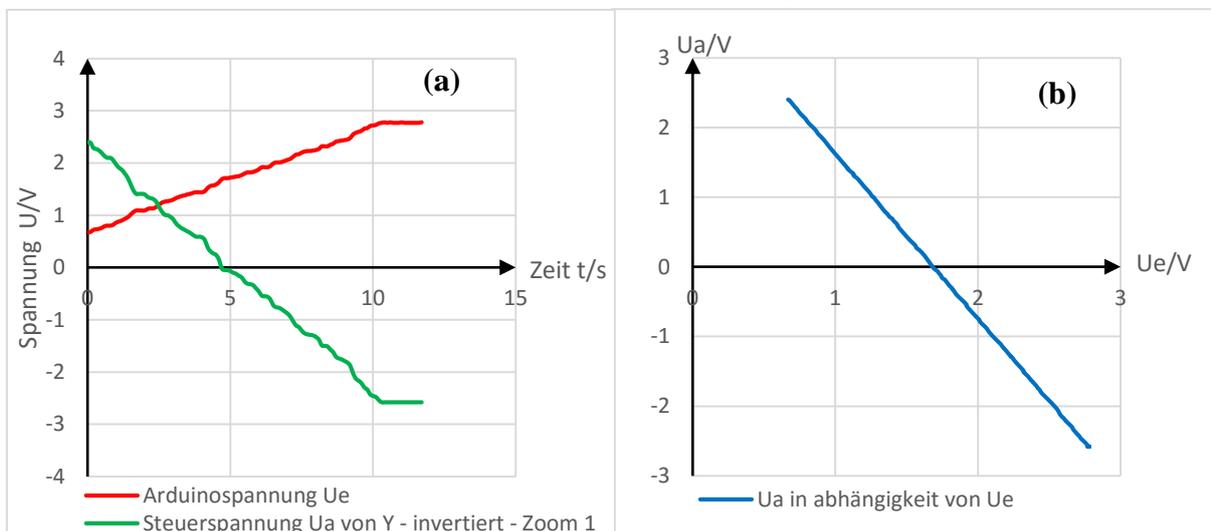


Diagramm 7: (a): Eingangsspannung (rot) und Ausgangsspannung invertiert (grün) in Y-Richtung Zoomstufe 1
 (b): Zusammenhang von Eingangs- zu invertierter Ausgangsspannung in Y-Richtung Zoomstufe 1

Insgesamt lässt sich an den Graphen erkennen, dass die Einstellungen bezüglich der Symmetrie und des maximalen Bildbereiches noch nicht perfekt sind. Jedoch hat dies keinen effektiven Einfluss auf die Messung, da lediglich der Bildausschnitt verkleinert oder verschoben, dabei aber nicht die Auflösung verschlechtert wird. Wichtig ist jedoch, den Bereich der Ausgangsspannung auf die maximale Spannweite von 0,55V bis 2,75V zu erweitern. Wie in den Graphen erkennbar, beginnt die Eingangsspannung bisher bei 0,67V und endet bei ca. 2,75V, wodurch ca. 0,12V als Auflösung verloren geht. Dies scheint zwar keine erhebliche Größe zu sein, entspricht jedoch einem Bildfenster von 223x233 Bildpunkten, also knapp 50000 Bildpunkten (vgl. Kapitel 5, Angaben zum Arduino).

5.3. Überprüfen der Nachregelung in Z-Richtung

Zur Überprüfung des Regelkreises, also der Z-Nachregelung der Messspitze, wurde ebenfalls eine Spannung am Eingang (das Tunnelstromäquivalent) des Regelkreises angelegt und gleichzeitig die an den Arduino Due übergebene Nachregelspannung gemessen.

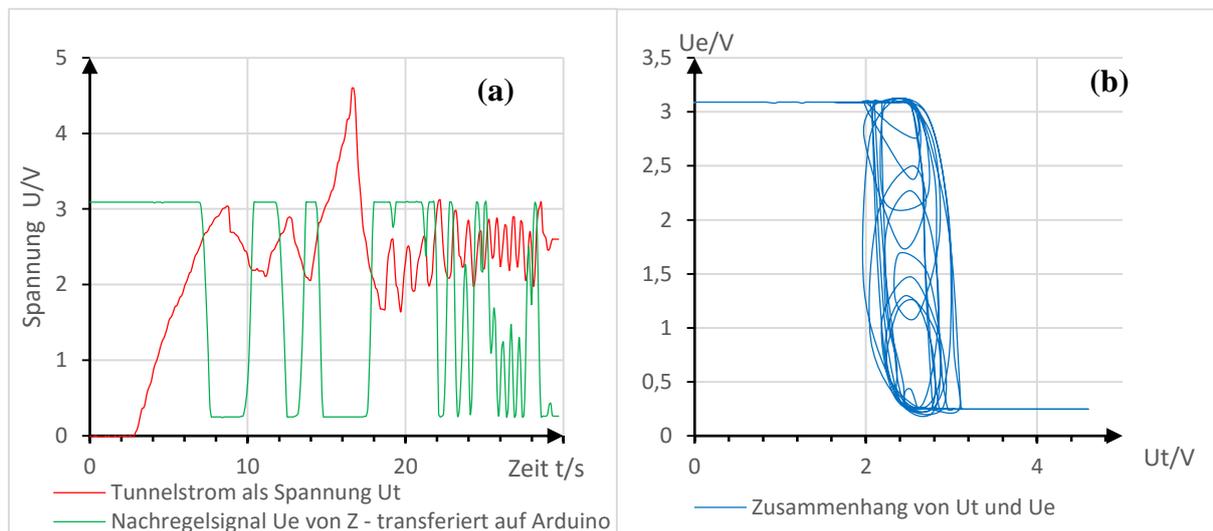


Diagramm 9: (a) Tunnelstrom als Spannung U_t (rot) & resultierendes Nachregelsignal transformiert für Arduino (grün)
(b) Tunnelstrom aufgetragen gegen transformiert Nachregelsignal

In Diagramm 9 (a) wird das Tunnelstromäquivalent simuliert und die daraus resultierende Nachregelspannung, welche am Arduino Due anliegt, dargestellt. Diagramm 9 (b) stellt den Zusammenhang der in Diagramm 9 (a) dargestellten Spannungsverläufe dar. Es ist erkennbar, dass das Nachregelsignal um einen festen Punkt pendelt, welcher der eingestellten Sollspannung von ca. 2,4V entspricht. Dies entspricht dem erwarteten Ergebnis.

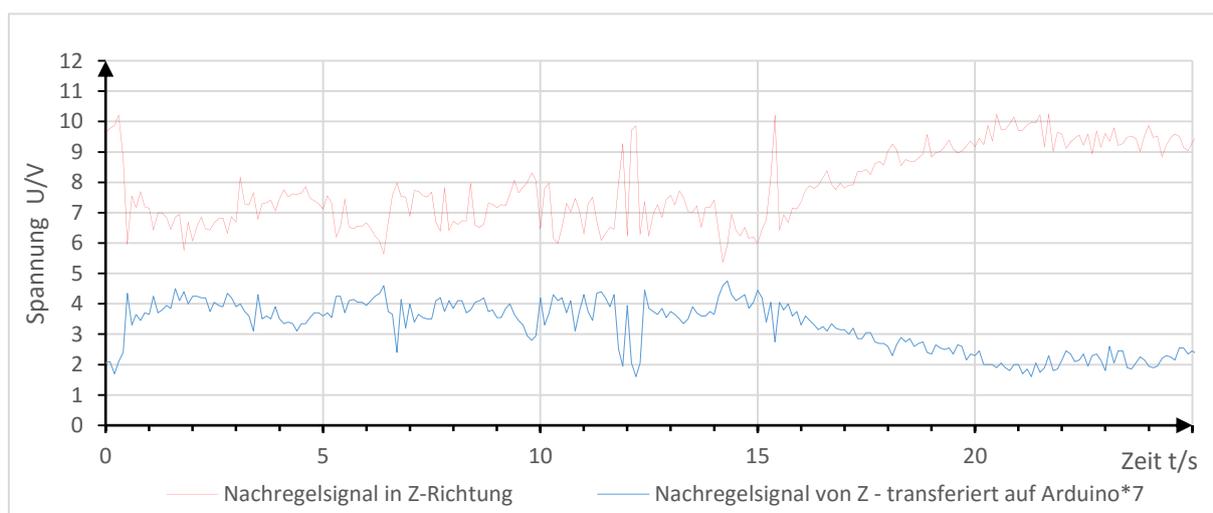


Diagramm 10: Tunnelstrom als Spannung U_t (rot) und die mit Faktor 7 multiplizierte, auf Arduino transformierte Spannung

Anmerkung: Die Y-Skala gilt nur zur Ablesung der Nachregelspannung, nicht für die verschobene Nachregelspannung, da diese mit dem Faktor 7 multipliziert wurde. Die Multiplikation wurde lediglich zur besseren Veranschaulichung der gezeigten Darstellung gewählt.

In Diagramm 10 wird die Spannung (rot) dargestellt, welche zur Nachregelung der Messspitze in Z-Richtung an den Piezo angelegt wird, ebenso das für den Arduino geschwächte und verschobene Signal (blau). Zur Veranschaulichung wurde für die Darstellung der am Arduino Due angelegten Spannung mit dem Faktor 7 multipliziert.

Anhand dem Diagramm ist ein sehr ähnlicher Verlauf beider Spannungen (die Invertierung ausgenommen) zu erkennen. Bei genauerer Betrachtung fällt jedoch auf, dass die Spannungsverläufe nicht den exakt selben Verlauf besitzen.

Durch ein Telefonat mit dem Entwickler Dr. Michael Hund des Cassy Lab 2 – Sensor von LD DIDACTIC, konnte die Vermutung bestätigt werden, dass die beiden Messeingänge zwar synchrone Messungen durchführen, diese jedoch bei Veränderungen jenseits der maximalen Abtastrate von einer Mikrosekunde, „*besser 2 Mikrosekunden*“ (Hund, 2016) keine exakt synchronen Messungen mehr durchführen kann.

Das bedeutet, dass die Abweichungen (Regelgeschwindigkeit lag laut Messung bei ca. 3,5ns) durch die zu niedrige Abtastrate zustande kommen dürfte.

6. Software

Bei Beginn der Arbeit, bestand bereits die Möglichkeit, die Messung des Tunnelstroms mit dem Rastertunnelmikroskop, ausgenommen die Annäherung der Spitze, vollständig vom Computer aus zu steuern. Hierzu wurde bereits ein Arduino Due als Kommunikationsschnittstelle, beziehungsweise als Messkarte verwendet. Außerdem bestand ein auf Python basierendes Programm zur Steuerung des STM. Zwar bot dies den Vorteil, dass alle gängigen Betriebssysteme unterstützt wurden, jedoch gab es nicht die Möglichkeit, eine ausführbare Datei, ohne Installation zusätzlicher Programme und Bibliotheken zu erstellen. Daher wurde eine benutzerfreundlichere Variante mittels Visual Basic erstellt. Zwar unterstützt dieses Programm lediglich Microsoft Windows Vista und nachfolgende (evtl. auch Windows XP), da dies allerdings in öffentlichen Einrichtungen, die meistgenutzte Plattform ist, ist dies kein allzu großer Nachteil. Das ursprüngliche Arduinoprogramm wurde von Jonas Dittmann erstellt. Dieses wurde lediglich um die Möglichkeit der Motorsteuerung erweitert. Außerdem wurde es minimal dahingehend abgeändert, dass die Kommunikation von Computer zu Arduino Due in 8bit Paketen vollzogen wird (vorher 16bit Pakete). Diese Veränderung wurde vorgenommen, da die Ausgabe von 16-bit Paketen an den Serial Port, seitens Visual Basic, sich als problematisch darstellte. Das mit Visual Basic geschriebene Programm kann sicherlich weitaus effizienter geschrieben werden, da die Programmierungsfähigkeiten jedoch nur auf autodidaktischer Aneignung basiert, stand die Funktionalität im Vordergrund.

6.1. Arduino Due - Software

Zur Kommunikation zwischen Computer und der Hauptplatine des STM wurde ein Arduino Due eingesetzt. Das Programm zur Messdatenerfassung, sowie Steuerung des STM war bereits vorhanden und wurde lediglich teilweise verändert. Dieses muss nur einmalig auf den Arduino Due mittels der Software „Arduino“ aufgespielt werden und verbleibt solange auf dem Gerät, bis eine neue Version aufgespielt wird. Zum besseren Verständnis, sowie zur Überarbeitung wurde die Arduino-Website zur Hilfe genommen (vgl. (Arduino, 2016)).

Das in Software „Arduino“ geschriebene Programm befindet sich im Anhang.

Das Programm ist grob folgendermaßen aufgebaut:

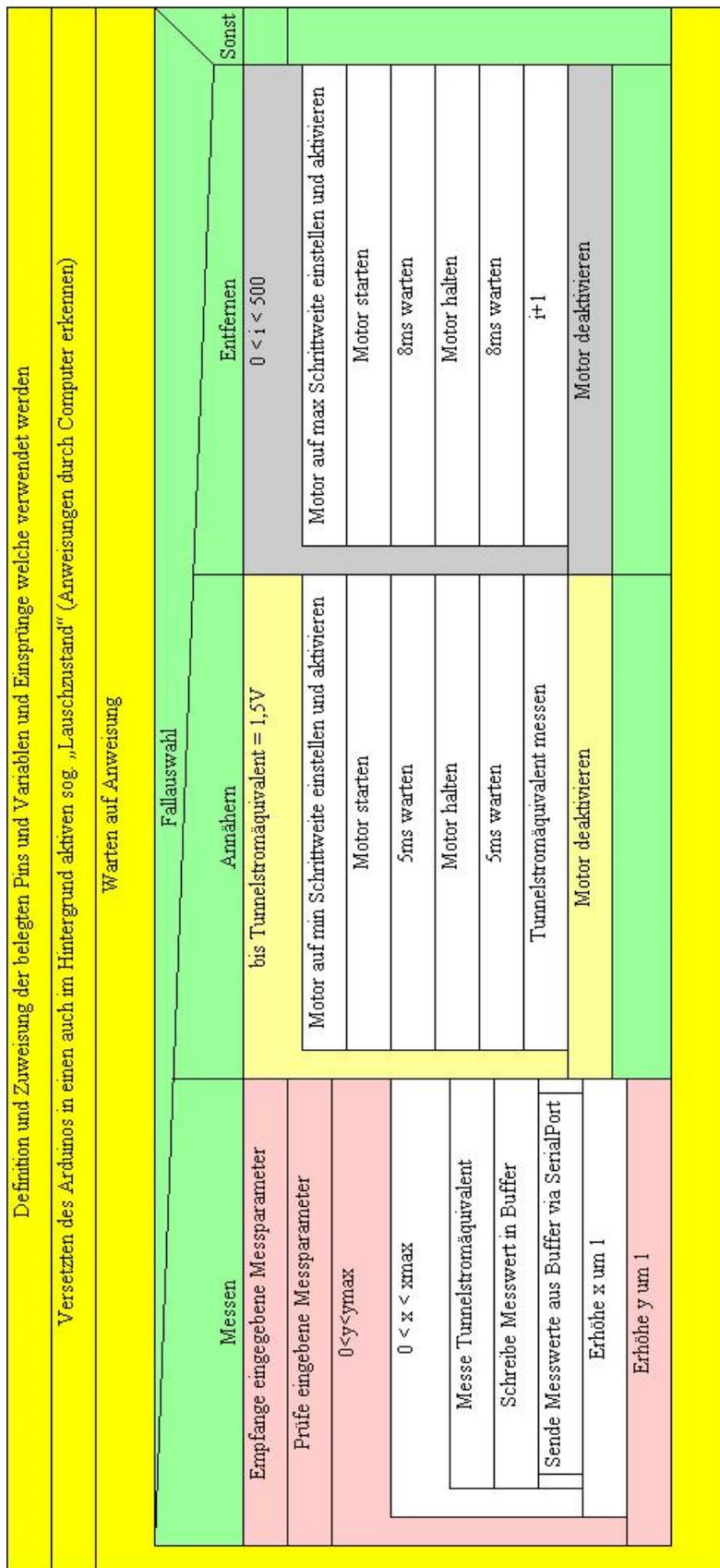


Abbildung 23: Schematischer Ablauf des Programmes auf dem Arduino

6.1.1. Spitzenannäherung/-entfernung

Ein neu eingefügter Bestandteil des Programmes ist eine automatische Annäherung der Messspitze an die Probe mittels Schrittmotor. Dies erfolgt durch den mit dem Arduino Due verbundene Schrittmotortreiber.

Erkennt der Arduino Due die Anweisung, die Messspitze der Probe zu nähern, so gibt er diese Anweisung an das Schrittmotormodul weiter. Unter anderem werden auch Informationen zur Geschwindigkeit des Motors gesendet. Um zu erfassen, wann die Spitze nahe genug an der Probe ist, wird zwischen jedem Schritt, welcher der Motor vollführt, eine Messung des Tunnelstroms vorgenommen.

Im Programmtext sieht dies dann wie folgt aus:

```

case AB: { //falls Anweisung Spitze nähern
  digitalWrite(MOTORENABLE,LOW); //Motor aktivieren (nicht starten)
  digitalWrite(MS1,HIGH); //Schrittweite definieren mit Pin MS1,2&3
  digitalWrite(MS2,HIGH); //Schrittweite definieren mit Pin MS1,2&3
  digitalWrite(MS3,HIGH); //Schrittweite definieren mit Pin MS1,2&3
  while (analogRead(AIN1)>2800){ // in Schleife bis Tunnelstrom ~1,5V
    digitalWrite(RICHTUNG, LOW); //Drehrichtung im Uhrzeigersinn
    digitalWrite(AUFAB, LOW); //Starte Motor
    delay (5); //warte 5ms
    digitalWrite(RICHTUNG, LOW); //Drehrichtung im Uhrzeigersinn
    digitalWrite(AUFAB, HIGH); //Stoppe Motor
    delay(5); // warte 5ms
  }
  digitalWrite(RICHTUNG, LOW); // Drehrichtung im Uhrzeigersinn
  digitalWrite(AUFAB, HIGH); //Stoppe Motor
  digitalWrite(MOTORENABLE,HIGH); //Motor deaktivieren
  break; } //Unterprogramm beenden

```

Quelltext Arduino 1: Ausschnitt- Spitze nähern

Zum Entfernen der Spitze von der Probe, wird ebenso der Schrittmotor genutzt. Hierzu fährt der Motor 500 Schritte gegen den Uhrzeigersinn, dies jedoch mit maximaler Schrittweite und ohne paralleles Messen des Tunnelstromäquivalents, wodurch die Entfernung weitaus schneller vonstattengeht.

6.1.2. Messung

Wird der Messvorgang angewiesen, so bearbeitet der Arduino Due mehrere Unterprogramme. Der in Abbildung 23 gezeigte Ablauf des Messens, spiegelt dabei nicht zwangsläufig die Unterteilung des Programmtextes dar, sondern den Ablauf der verschiedenen Unterprogramme in Kombination mit der Computersoftware.

Programmintern (im Arduino Due) unterteilt sich die Messung und alles Dazugehörige in folgende Unterprogramme:

Für das gesamte Programm gilt:

In steht für Zeile bzw. y-Richtung, px steht für Spalte bzw. x-Richtung

- **Setupramp**

Es werden die vom Computer ausgegebenen Messparameter eingelesen:

```

px_min    = ((Serial.read() * 256)); //X-Minimalwert = eingelesener Wert *256
px_neutral = (((Serial.read() + 1) * 256)); //X-Startwert = (eingelesener Wert+1) *256
px_max    = (((Serial.read() + 1) * 256)); //X-Endwert = (eingelesener Wert+1) *256
px_increment = (1 << (Serial.read())); //X-Schrittweite = „1“ linksverschoben um
//eingelesenen Wert
px_sawtooth = Serial.read(); //X-Sägezahn = eingelesener Wert (ja/nein)

```

Quelltext Arduino 2: Ausschnitt - Einlesen der Messparameter

Anzumerken ist, dass die Multiplikation der eingelesenen Daten mit 256, deshalb vorgenommen wurde, da so auf eine 16-Bit Übertragung verzichtet werden konnte.

Der Linksshift der 1 wurde deshalb vorgenommen, da dies in Kombination mit der Computersoftware den Vorteil bietet, den Regler zur Schrittweite, in der Abstufung 0,1...6, laufen zu lassen, wodurch arduinointern Schrittweiten von $2^0, 2^1, \dots 2^6$ folgen. Die festen Schrittweiten, beziehungsweise die Potenzen von 2, wurden deshalb gewählt, um zu gewährleisten, dass das so entstehende Raster in Ganzzahlige, äquidistante Bereiche unterteilt wird.

Das Einlesen der Y-Messparameter verläuft analog.

Anschließend werden die Werte auf ihre Plausibilität überprüft. Hierzu sind programm-intern Rahmendaten festgelegt, welche eingehalten werden müssen.

```

if (px_max > 4096)           return false;
if (px_neutral > 4096)      return false;
if (px_max <= px_min)       return false;
if (px_increment > (px_max - px_min)) return false;
if (px_increment == 0)      return false;

```

Quelltext Arduino 3: Ausschnitt - Überprüfen der Messparameter in x-Richtung

Entspricht ein eingegebener Wert nicht den obigen Bedingungen, so werden die eingelesenen Werte verworfen. Dieser Fall kann jedoch nicht auftreten, da seitens der Computersoftware schon bei Eingabe der Daten eine Überprüfung stattfindet. Die Überprüfung wurde jedoch als Redundanz vorerst beibehalten.

- **Start**

Startet die Messung, so beginnt dieses im Normalfall (in der Computersoftware so festgelegte) bei den eingegebenen Minimalwerten von px und ln .

Bei $px = px_{min}$ und $ln = ln_{min}$ wird eine Messung vollzogen. Anschließend wird px um eine Schrittweite erhöht und wieder eine Messung durchgeführt. Dies wird solange wiederholt, bis $px = px_{max}$. Nun wird ln um eine Schrittweite erhöht, $px = px_{min}$ gesetzt und gemessen. Die sequentielle Erhöhung von px und die Messungen werden wieder durchgeführt.

Dieser Vorgang wird solange wiederholt, bis $ln = ln_{max}$ und $px = px_{max}$ ist. Dann sind die Messungen abgeschlossen. Die Messergebnisse werden parallel zum Messvorgang via Serial Port verschickt.

- **Stopp**

Empfängt der Arduino Due das Stoppsignal, so beendet er jegliche Vorgänge. Befindet er sich zum Zeitpunkt des Signalempfangs in einer Messung, so wird aber bis zum Abschließen der Einzelmessung, nicht jedoch bis zum vollständigen Abschließen der gesamten Messung, gewartet.

6.2. Computer-Software

Die Computersoftware wurde mit Visual Studio 2013, in der Programmiersprache Visual Basic .Net erstellt. Somit hat man die Möglichkeit, eine ausführbare Datei zu erstellen, welche ohne Installationen, mit jedem „aktuelleren“ (Windows Vista und folgende) Windows-Betriebssystem kompatibel ist. Getestet wurde dies an mehreren Rechnern, kann jedoch aufgrund von eventuell veralteter Net Framework Versionen nicht gesichert werden.

Folgende Systemvoraussetzungen sind für den Betrieb und die Nutzung erforderlich:

- NetFramework Ver. 4.5
- Installierter Treiber des Arduino Due, sowie angeschlossen an USB-Port

Des Weiteren ist zum Betrachten der erstellten 2D-Bilder als 3D-Grafik Java notwendig.

Das in Visual Studio 13 geschriebene Programm kann dem Anhang entnommen werden. Als Programmierungshilfe wurde die Online-Dokumentation zu Visual Basic .Net genutzt (vgl. (Microsoft Corporation, 2016)).

6.2.1. Oberfläche

Die Oberfläche wurde dahingehend designt, dass eine neutrale Optik entsteht, sowie nur die häufig genutzten Einstellungen sich auf dem Hauptfenster befinden. Ziel war es, das Programm möglichst selbsterklärend darzustellen.

So wird bei der Erstausführung, beziehungsweise bei nicht-angeschlossenem Arduino Due, direkt eine Meldung wiedergegeben, welche den Benutzer dazu auffordert, die Einstellungen zu überprüfen.

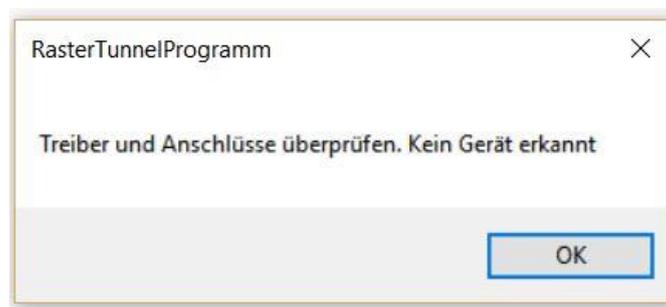


Abbildung 24: Warnmeldung über nicht erkanntes Gerät

Zwar sind Standardwerte hinterlegt, diese beinhalten jedoch nicht die Einstellung der korrekten Kommunikationsschnittstelle (COM-Port). Ein automatisches Erkennen, welchen COM-Port der Arduino Due nutzt, konnte bisher nicht erfolgreich eingebunden werden, weshalb ein manuelles Einstellen der Portbelegung notwendig ist. Daher wird man nach Bestätigung der Meldung, direkt zum „Einstellungen“-Fenster weitergeleitet.

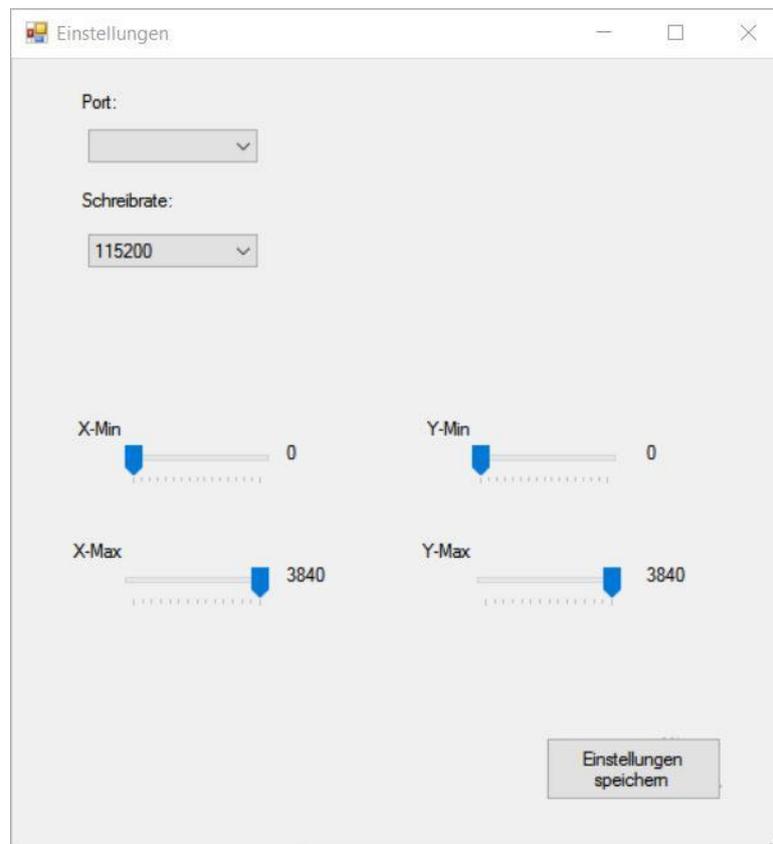


Abbildung 25: Einstellungsfenster des Raster Tunnel Programmes

Innerhalb dieses Fensters befinden sich mehrere Einstellungen. Unter dem Punkt *Port* muss der zuständige COM-Port des Arduino Due ausgewählt werden. Hierfür ist es von Vorteil, wenn, ausgenommen Maus oder Tastatur, keine weiteren Geräte an dem jeweiligen Computer angeschlossen sind, da so lediglich ein Port zur Auswahl steht, welcher den Arduino repräsentiert. Die Baudrate wurde standardmäßig auf 115200 Bit/s eingestellt. Zur weiteren Auswahl steht 9600Bit/s zur Verfügung. Die niedrigere Baudrate kann gewählt werden, falls es zu, zu vielen Übertragungsfehlern kommen sollte, dies bedeutet jedoch, dass die 12-fach Zeit zur Übertragung von Daten benötigt wird. Unter X/Y-Min/Max kann die Größe des zu scannenden Rasters eingestellt werden. Ist das Minimum von X oder Y größer/gleich dem entsprechenden Maximum, so wird eine Fehlermeldung ausgegeben und ein Speichern der Einstellungen, sowie Schließen des Einstellungsfensters verhindert.



Abbildung 26: Warnung bei ungültigen X/Y Randwerten

Wurden korrekte Werte eingetragen, so können die Einstellungen gespeichert werden. Diese werden, sofern keine Änderungen mehr vorgenommen werden, für den nächsten Start beibehalten. Dabei werden jedoch alle Einstellungen unabhängig voneinander gespeichert. Das heißt, sollte sich der COM-Port ändern, wird zwar zur erneuten Auswahl des Ports aufgefordert, jedoch bleiben andere Einstellungen unberührt.

Die Hauptoberfläche des Programmes sieht letztendlich wie folgt aus:

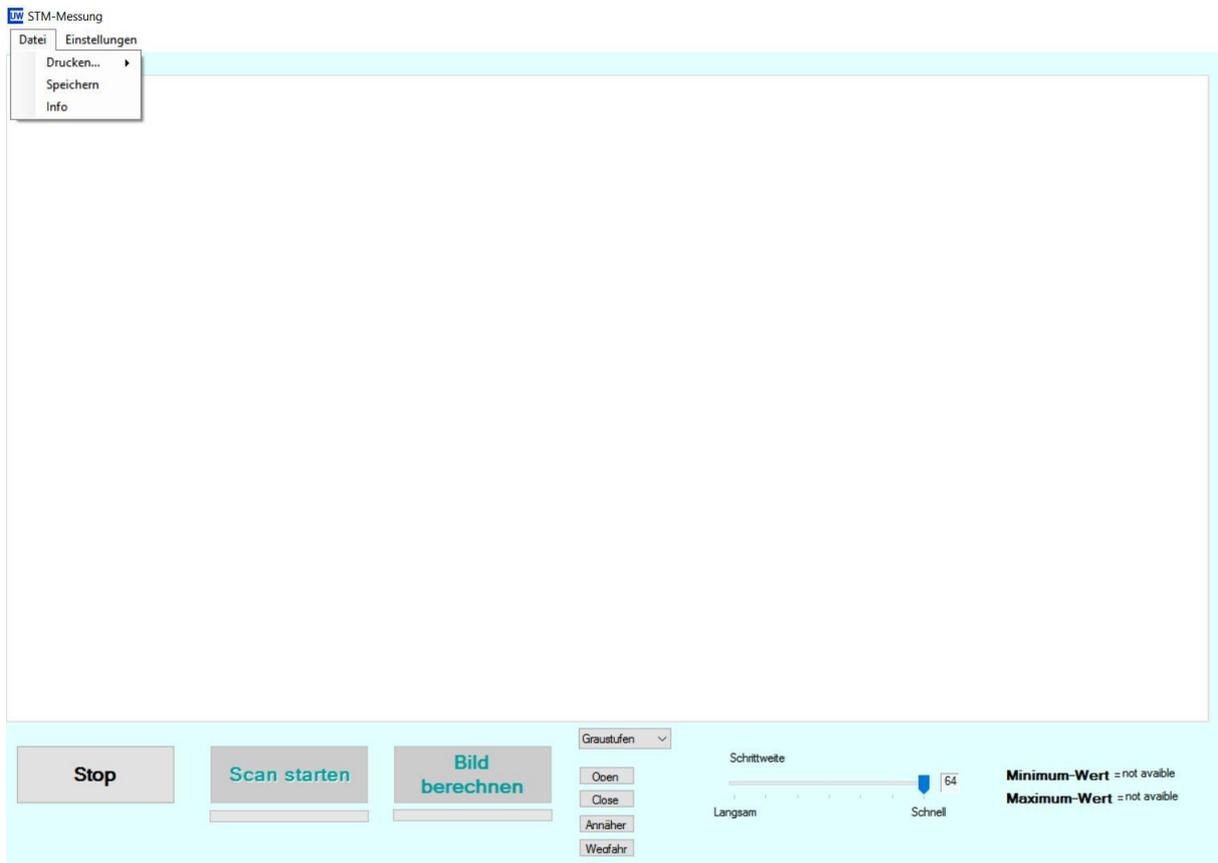


Abbildung 27: STM-Messprogramm Oberfläche

Ein hochauflösendes Bild der Oberfläche kann dem Anhang entnommen werden.

Die unter „Einstellungen“, als Bildrand zu wählenden Werte, gehen von 0 bis 3840, in 256er Schritten, sowie die im Hauptfenster zu wählende Schrittweite beträgt $2^0, 2^1 \dots 2^6$.

Daraus resultiert ein Bild, welches in jedem Fall teilbar durch die Schrittweite ist.

Als Beispiel:

$$x_{min} = 512, \quad x_{max} = 1024, \quad y_{min} = 0, \quad y_{max} = 512, \quad \text{Schrittweite} = 8$$

Dementsprechend:

$$\text{Anzahl Messwerte/Bildpunkte} = \frac{(y_{max} - y_{min}) * (x_{max} - x_{min})}{\text{Schrittweite}^2} = 496 \quad (15)$$

6.2.2. Kommunikation

Der Computer kommuniziert mit dem Arduino Due via COM-Port. Dieser muss vor der Kommunikation geöffnet und am Ende wieder geschlossen werden. Dieser Vorgang wurde zu Beginn jeder Messung automatisch vollzogen, zeigte jedoch in der Praxis das Problem, dass nach schließen des COM-Ports, der Arduino Due in einen „Ruhezustand“ versetzt wird, wodurch sämtliche Ausgänge in regelmäßigen Abständen einen kurzen Spannungsstoß erhalten. Dies führt zu dem Problem, dass die Messspitze in sehr kurzer Zeit maximal ausgelenkt wird und so die Regelelektronik zur Höhenregulierung nicht schnell genug reagieren kann, was zur Folge hat, dass die Spitze gegen die Probe stößt, wodurch sie unbrauchbar wird.

Abhilfe schafft das Offenlassen des COM-Ports während der Nutzungszeit. Daher ist vorerst ein manuelles Öffnen und Schließen des Ports, mittels Button, im Programm implementiert.

Sämtliche Daten zwischen Computer und Arduino Due werden als 1Byte Pakete ausgetauscht. Dabei gehen folgende Daten vom Programm an den Arduino Due:

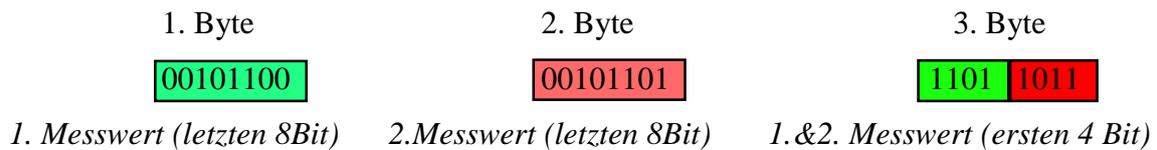
- X/Y- Randwerte
- Schrittweite
- Startpunkt der Messung
- Nutzung der Sägezahnfunktion (immer deaktiviert)

Das Rastern mit dem Verlauf einer Sägezahnfunktion ist eine, im Arduino-Code schon vorhandene, Abwandlung, welche vorerst nicht entfernt wurde, jedoch nicht genutzt wird, da eine Doppelbildaufnahme Aufschluss über mögliche Messfehler (z.B. zu langsame Nachregelung) geben kann.

Der Arduino Due dagegen sendet seine Daten in spezieller Form. Da jeder Messwert eine Speichergröße von 12 Bit besitzt, eine Übertragung jedoch nur in 8Bit Paketen vorgenommen werden kann, werden die Daten aufgespalten, um die höchstmögliche Bandbreite zu nutzen.

Seiten des Arduino Due werden zwei Messwerte, also 24 Bit, als 3 x 1 Byte Pakete gesendet. Um sicherzustellen, dass immer zusammenhängende Daten empfangen werden, wird eine Art „Prüfsumme“ eingesetzt. So werden die gesamten Messdaten in einer Gruppe von 16 Byte verschickt, wobei jedes erste Byte der Gruppe den Dezimalwert 66 besitzt und als Startbyte der Gruppe gilt. Darauf folgen fünf 3-Byte Gruppen, welche 10 Messwerte enthalten.

Als Beispiel sieht eine 3-Byte Gruppe im Binären wie folgt aus:



Durch Zusammensetzen der 3 Byte, folgen die beiden 12-Bit Messwerte:



Die so empfangenen Messwerte werden vorerst der Reihe nach in einem Buffer gespeichert.

6.2.3. Messung starten

Vor Messbeginn, muss der COM-Port geöffnet werden. Dies geschieht durch einen einfachen Klick auf den Button „Open“ (siehe Abbildung 27). Anschließend kann die Messspitze mittels Klick auf den Button „Annähern“, automatisch der Probe genähert werden. Das Programm sendet durch den COM-Port ein Signal, das arduinointern das Unterprogramm zur Spitzennäherung gestartet wird (siehe Kapitel 6.1.1).

Ist die Messspitze an die Probe herangefahren, so kann durch Klick auf „Scan starten“, mit der eigentlichen Messung begonnen werden. Die in Kapitel 6.2.2 genannten Messparameter werden übermittelt. Die genauen Werte entnimmt das Programm den Schieberegler-Positionen.

Anhand der Randwerte (X/Y-min/max) und der Schrittweite kann die Anzahl der zu empfangenden Messwerte und damit Bytes berechnet werden.

$High = CInt(2 * (((hmax - hmin) / (schrittweite))))$ //Berechnen der Bildhöhe (Anzahl Punkte)

$Length = CInt(((lmax - lmin) / (schrittweite)))$ //Berechnen der Bildbreite (Anzahl Punkte)

$h = CInt(((Length * High) / 5) * 8) + hmax$ //Berechnen der zu empfangenden Bytes

*Quelltext Visual Basic 1: Ausschnitt Unterprogramm Scannen -
Berechnung der zu empfangenden Bytes*

Diese Anzahl ist notwendig, um das Ende des Datenempfangs zu bestimmen.

Zusätzlich wird diese Anzahl genutzt, um parallel zur Messung einen Fortschrittsbalken anzeigen zu können.

ProgressBar.Maximum = h + 5 //Maximum des Fortschrittbalkens „Scan starten“ festlegen

ProgressBar1.Maximum = h + 5//Maximum des Fortschrittbalkens „Bild berechnen“ festlegen

*Quelltext Visual Basic 2: Ausschnitt Unterprogramm Scannen -
Einstellung des Fortschrittbalkens*

Im Folgenden werden die empfangenen Daten eingelesen und in einem Buffer gespeichert.

Zu Beginn des Unterprogramms *Scannen* wurde die Variable $n = 0$ gesetzt.

```

Do While n < h                // Schleife solange n kleiner h ist
    buffer(n) = SerialPort1.ReadByte() // gelesenen Bytes in Buffer(n) speichern
    buffer(n + 1) = SerialPort1.ReadByte() //....in Buffer(n+1) speichern
    buffer(n + 2) = SerialPort1.ReadByte() //....in Buffer(n+2) speichern
    buffer(n + 3) = SerialPort1.ReadByte() //....in Buffer(n+3) speichern
    buffer(n + 4) = SerialPort1.ReadByte() //....in Buffer(n+4) speichern
    n += 5                    // n um 5 erhöhen
    ProgressBar1.Value = n    // Fortschrittsbalken auf Wert n setzten
    If stopbit = 1 Then      // Falls „Stop“ Button gedrückt wurde
        stopbit = 0        // “Stop“- Buttonwert resettet
        SerialPort1.Write(Halt, 0, 1) // Haltbefehl an Arduino senden
        MsgBox("Messung abgebrochen") // Warnmeldung ausgeben
        GoTo ende          // Ausprung aus Unterprogramm
    End If                  // Beenden der If-Schleife
Loop                        // Schleife beenden falls n≥h
m = n                       // m wird auf erreichten n wert gesetzt
n = 0                       // n wird auf 0 gesetzt

```

*Quelltext Visual Basic 3: Ausschnitt Unterprogramm Scannen -
Einlesen der zu empfangenden Bytes*

Anschließend müssen die vom Arduino Due nach Kapitel 6.2.2 empfangenen Daten wieder zusammengesetzt werden.

Zu Beginn des Unterprogramms *Scannen* wurde die Variable *counter* = 0 gesetzt.

```

Do While n < m           // Führe Schleifen so lange aus wie n<m
  Do Until buffer(n) = 66 // Führe Schleife aus bis Buffer = Start-Byte einer
                        // Gruppe (vgl. Kapitel 6.2.2)
    n += 1              // Erhöhe n um eins
  Loop                  // gehe an Start der Schleife
  datbyte = buffer(n)   // Schreibe aktueller Bufferwert (66) in Datbyte
  packets = ((datbyte And 3) + 1) * 5 // Speichern der zu erwartenden Pakete
  channel = (datbyte And 12) // Reservefunktion
  command = (datbyte And 240) // Reservefunktion
  p = 1                 // Variable p wird auf eins gesetzt
  n += 1                // Variable n wird um eins erhöht
  Do While p < packets And (n - 3) < m // Bearbeite Schleife solange Pakete erwart-
  // et werden und min. 3 Bytes unverarbeitet
    If i = 1 Then       // Führe Schleife aus falls i=1
      a = buffer(n)     // In a wird akt. Bufferbyte geschrieben
      i = i + 1         // erhöhe i um eins
      p += 1            // erhöhe p um eins
      n += 1            // erhöhe n um eins
    End If              // Ende der Schleife falls Bedingung wahr
    If i = 2 Then       // Führe Schleife aus falls i=2
      b = buffer(n)     // In b wird akt. Bufferbyte geschrieben
      i = i + 1         // erhöhe i um eins
      p += 1            // erhöhe p um eins
      n += 1            // erhöhe n um eins
    End If              // Ende der Schleife falls Bedingung wahr
    If i = 3 Then       // Führe Schleife aus falls i=3
      c = buffer(n)     // In c wird akt. Bufferbyte geschrieben, c
                        // enthält dabei Bits die zu sowohl a als auch
                        // b gehören, siehe Kapitel 6.2.2

```

```

a = a Or ((c And 15) << 8)           // Zusammensetzen der Bits aus a & c nach
                                     // Kapitel 6.2.2 – c=Mischbyte

b = b Or ((c >> 4) << 8)           // Zusammensetzen der Bits aus b & c nach
                                     // Kapitel 6.2.2

counter = counter + 1               // Erhöhe Counter um 1

daten(counter) = (a)                // Schreibe in Datenarray, Daten aus a

counter = counter + 1               // Erhöhe Counter um 1

daten(counter) = (b)                // Schreibe in Datenarray, Daten aus b

i = 1                                // i wird auf eins gesetzt

p += 1                               // p wird um eins erhöht

n += 1                               // n wird um eins erhöht

End If                               // Ende der Schleife falls Bedingung wahr

Loop                                 // gehe an Start der Schleife

Loop                                 // gehe an Start der Schleife

SerialPort1.Write(Halt, 0, 1)       // gebe Arduino Stopp-Signal

```

*Quelltext Visual Basic 4: Ausschnitt Unterprogramm Scannen -
Eingelesene Bytes werden wieder zu Messwerten zusammengesetzt*

Wurden alle Daten empfangen und als 12bit – Messwerte im Datenarray gespeichert, so wird die Meldung „Messung beendet!“ ausgegeben.

Hat man beispielsweise unbeabsichtigt eine zu kleine Schrittweite gewählt, so kann die Messung mittels „Stop“ abgebrochen werden. Alle bis dahin gesammelten Daten werden verworfen und die Meldung „Messung abgebrochen“ ausgegeben.

6.2.4. Messung visualisieren

Nachdem die Messung beendet ist, kann mittels Klick auf „Bild berechnen“ die gemessenen Werte in einem zweifarbigen ARGB-Bild visualisiert werden, welches der vorherigen Farbwahl entspricht. Hierzu wird programmintern vorerst der kleinste, sowie der größte Messwert und anhand dieser die *Spannweite* S der Daten bestimmt.

Anschließend wird jeder Messwert in Scanreihenfolge, also gleichbedeutend der Reihenfolge des Datenempfangs, als Bildpunkt in eine Grafik eingefügt, wobei die Farbe des Bildpunktes dem Wert des Messwertes entspricht und die Position des Messwertes, auf die des Bildes übertragen wird. Das bedeutet, das entstehende Bild entspricht dem gerasterten Probenausschnitt.

Dabei wird die Farbe wie folgt berechnet:

Gegeben sei:

$$S = 2027, \text{Min. Wert} = 1034, \text{Max. Wert} = 3061, \text{Bsp. Wert} = 2803$$

Gewählt wurde ein Graustufenbild, das würde bei einem ARGB-Bild bedeuten (vgl. (Microsoft Corporation, 2016)), Alpha =255 und $R = G = B = m_{x,y}$ (x- sowie y-Koordinaten entsprechend dem jeweiligen Messwert).

Zur Bestimmung der *Farbintensivität* m wird von allen Messwerten der vorher ermittelte, kleinste Wert abgezogen, um diesen farblich gesehen, als „Nullpunkt“ zu definieren. Anschließend werden alle Werte durch den Quotienten der Spannbreite und 240 geteilt. Theoretisch kann anstatt 240, 255 verwendet werden, da diese Zahl dem Maximalwert einer Farbe entspricht (Beschränkung liegt bei 8 Bit je Farbkomponente), zeigte sich jedoch in der Praxis, aufgrund von bisher nicht lösbaren Rundungsfehlern, als problematisch.

Daher greift für die Farbberechnung eines jeden Messwerts folgende Formel:

$$m_{x,y} = \left\lfloor \frac{(\text{aktueller Messwert} - \text{Min. Wert})}{\frac{S}{240}} \right\rfloor \quad (16)$$

Wobei $\lfloor * \rfloor$ dem kaufmännischen Runden entspricht.

Damit folgt für jeden Bildpunkt p eines Graustufenbildes $p(x,y) = (255, m_{x,y}, m_{x,y}, m_{x,y})$

Dieser Prozess wird für jeden einzelnen Punkt wiederholt bis alle Messwerte verarbeitet wurden. Diese Formel garantiert, dass je nach Messung, die Spannweite der Messwerte, maximal auf das verwendete Farbspektrum aufgeteilt wird.

6.2.1. Speichern des Bildes/Ausdrucken und 3D-Ansicht

Anschließend wurde in das STM-Messprogramm die Möglichkeit eingebettet, das so entstandene Bild, unter dem Reiter „Datei“, als *.bmp Grafik abzuspeichern oder direkt zu drucken. Das Sichern des Bildes ist notwendig, um die 2D als eine 3D-Grafik zu visualisieren. Dies wird mit dem Freeware Programm *ImageJ*, welches auf Java basiert, umgesetzt, daher ist Java auf dem System vorausgesetzt.

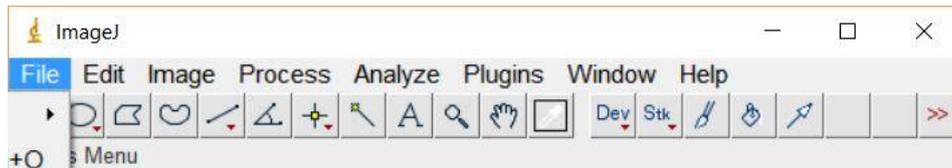


Abbildung 28: ImageJ – Startfenster

Mittels Reiter „File“ → „Open“ kann anschließend das vorher entstandene Bild geöffnet werden. Nun wird unter dem Reiter „Plugins“ → „3D“ → „Interactive 3D Surface Plot“ eine interaktive 3D Ansicht gestartet. Dieses Plugin nutzt zur Erstellung der Z-Achsenwerte, Kontrastunterschiede innerhalb des Bildes, dabei bleibt jedoch die Position innerhalb des Bildes (XY-Wert) unverändert.

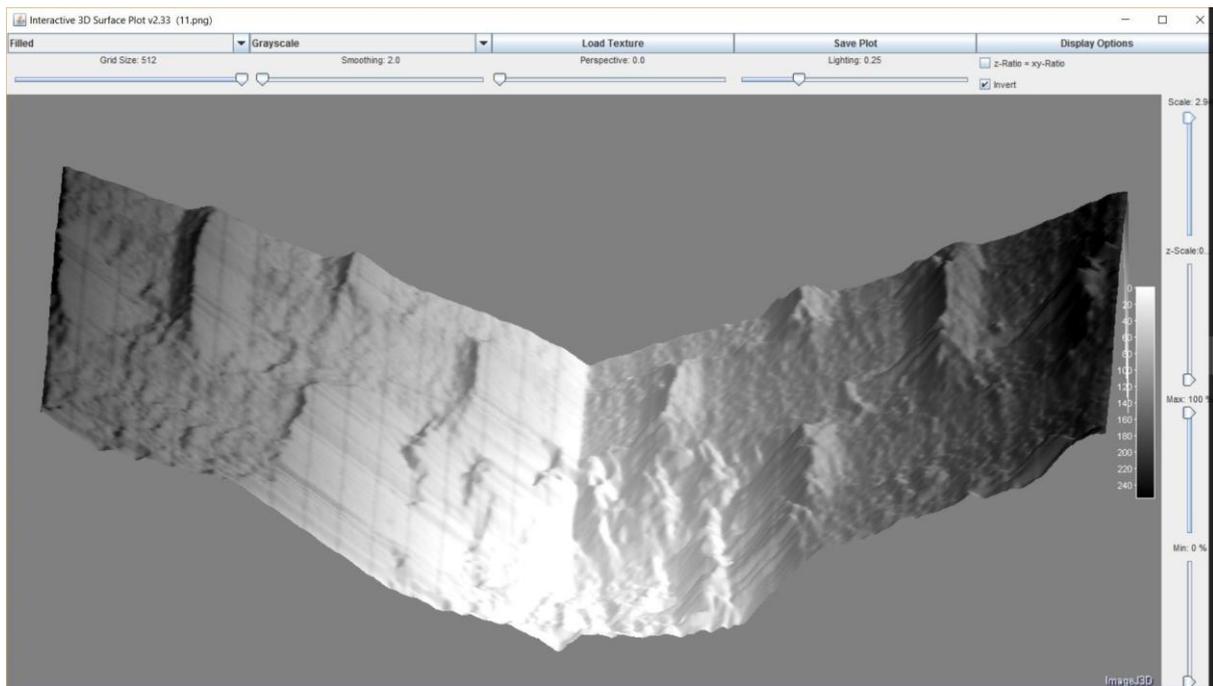


Abbildung 29: Image J Beispiel eines 3D Plots (Aufnahmen eines DVD-Masters vom 17.10.2015)

Anhand der Schieberegler, können nun Z-Minimum, Z-Maximum, Z-Skalierung, Gesamtskalierung, Perspektive, Reflektion und Glättung sowie einige weitere Parameter eingestellt werden. Hier empfiehlt es sich, die Wirkung einzelner Regler zu testen. Außerdem kann die Grafik mittels Mauszeiger in alle 3 Raumrichtungen gedreht werden. Gespeichert wird die grafische Darstellung anschließend mit Klick auf „Save Plot“.

7. Nutzung des STM im Schülerlabor

7.1. Allgemeine Erklärung

Das Rastertunnelmikroskop und dessen Einzelteile, sowie die Einzelschritte bis zu einer Messung umfassen ein sehr breites Themengebiet. So könnte je nach Schulart, der Zusammenbau des Gerätes ebenfalls durch Schüler stattfinden, dafür wäre ein technischer Zweig ideal. Es bietet die Möglichkeit, sowohl in Schulen als auch außerhalb, in einem Schülerlabor wie es die Universität Würzburg anbietet, eingesetzt zu werden. Da die Einzelkomponenten sehr günstig sind, lohnt es sich auch für Schulen, welche über nur ein geringes Budget verfügen.

Für die Bedienung des STM ist keinerlei physikalisches Vorwissen notwendig, weshalb Messungen auch von leistungsschwachen Schülern, oder Schülern unterer Klassenstufen im Sinne eines außerlehrplanmäßigen Projekttages durchgeführt werden kann. Dies bietet den Vorteil, selbst uninteressierte Schüler für den physikalischen Hintergrund, und somit für das Fach Physik, zu motivieren. Bei Schülern der Oberstufe hingegen könnte die Thematik des Tunneleffekts vertieft behandelt werden. Empfehlenswert wäre es jedoch, die Messungen nicht als Demonstrationsversuch, sondern als Schülerversuch mit einem Lehrer als Berater, welcher bei Problemen zur Seite steht, durchzuführen. Da sowohl die Messungen, als auch die Vorbereitungen einige Zeit benötigen, sollte hierfür ein ganzer Schultag vorgesehen werden.

Da mit einem Rastertunnelmikroskop, Topographien im Mikro-, bis hin zum Nanometerbereich sichtbar gemacht werden können, bietet das STM eine wunderbare Möglichkeit, Schülern mit dem Scan eines DVD-Masters und eines CD-Masters die Unterschiede, sowie Gemeinsamkeiten der beiden Datenträger darzulegen. Dies geschieht meist mit großer Faszination. Der Schülerversuch als Methodik, bietet den Vorteil, dass der Lerneffekt und der für die Motivation sehr wichtige „Wow-Effekt“ viel stärker erfahren wird. Wird schließlich noch der Zusammenbau des STM mit den letztendlichen Messdurchführungen gekoppelt, so wird dieser Effekt nochmals verstärkt.

Der Schüler erkennt, mit welchen einfachen Mitteln ein sonst hochkompliziertes, sowie sehr teures Gerät, „nachgebaut“ werden kann. In solch einer Projektphase sollte jedoch nicht nur die Gegenwartsbedeutung, sondern ebenso die Zukunftsbedeutung eines Rastertunnelmikroskops, als auch die des Tunneleffekts, verdeutlicht werden. So könnte auf die Möglichkeit der Nanomanipulation mittels Rastertunnelmikroskop eingegangen werden, bei der mittels der Messspitze, nicht nur Strukturen erfasst, sondern auch verändert werden können (vgl. (Chen, 2009 S. 2)).

So können einzelne Atome mithilfe eines Rastertunnelmikroskops entfernt oder an bestimmte Orte verlagert werden, was ein möglicher Meilenstein hinsichtlich der Speichertechnologie sein könnte. So gelang es zuletzt Forschern der Technischen Universität in Delft, Daten auf einem Chloratomgitter, welches sich auf einer Kupferoberfläche befand, zu schreiben (vgl. (Lölfken, 2016)). Andererseits kann auf den Nutzen des Tunneleffekts im Alltag eingegangen werden, welcher beispielsweise bei QTC-Material (*“Quantum Tunnelling Composite - Material”*) ausgenutzt wird. Dieses Material bietet die Möglichkeit, Drucksensoren zu bauen, welche vollständig neue Dimensionen eröffnen, da sich der Widerstandswert durch den Tunneleffekt, bei Druck, im Vergleich zu unbelastetem Sensor, um bis zu 10^{12} erhöhen kann (vgl. (Peratech Holdco Limited, 2016)). Dies ist vor allem für Elektroartikel, welche Touch-Eingabe nutzen, interessant, da so hoch sensible Touch-Elemente gebaut werden können, die selbst mit Handschuhen bedienbar sind, ohne auf resistive Touch-Elemente (vgl. (Ansgar, et al., 2007)) zurückgreifen zu müssen.

Ein Beispiel, in welchem der Tunneleffekt eher Probleme bereitet, ist der Bau von Prozessoren. Denn die Leistungssteigerung der Prozessoren begründet sich bisher verstärkt in der Verkleinerung der einzelnen Transistoren. Werden jedoch die Barrieren der dieser zu klein dimensioniert, so treten ungewollte Tunneleffekte auf, ohne, dass diese kontrollierbar sind (vgl. (Gehring, 2003)).

7.2. Bedienungsanleitung zu selbstständigen Verwendung

Zur eigenständigen Verwendung des STM wird von einem fertigen Versuchsaufbau ausgegangen. Das bedeutet, alles ist bereits installiert und angeschlossen, sowie der COM-Port des Arduino Due bereits im Programm gewählt ist. Je nach Befähigung der durchführenden Person(en), kann Schritt 2 von der aufsichtführenden Person erledigt werden.

Schritt 1 – Vorbereitungen:

- Für ruhiges Umfeld sorgen
- Zu untersuchende Probe bereitlegen
- STM-Messungsprogramm starten
- Grobgewindeschraube der Scaneinheit bis zum Anschlag im Uhrzeigersinn drehen
- Feingewindeschraube, auf etwa die Hälfte der maximalen Tiefe einschrauben (Motor währenddessen entkoppeln)

Schritt 2 – Spitze herstellen:

- Mit Latexhandschuhen ein ca. 1,5cm langes Stück Wolframdraht (Dicke 0,2mm) abschneiden, leicht anschleifen, durch destilliertes Wasser ziehen und an einen der bereitgestellten Kupferdrahtstäbe (Dicke 2mm) anlöten. Der Wolframdraht sollte nicht mit der bloßen Hand berührt werden. Das Ende des Wolfram-Drahtes in einen kurzen Abschnitt eines sehr dünnen Kabels stecken, sodass noch etwa 7mm des Wolfram-Drahtes als Zwischenstück sichtbar ist (siehe Bild 1).



Bild 1: Messspitze bereit zum Ätzen

Das Ende des Wolfram-Drahtes in einen kurzen Abschnitt eines sehr dünnen Kabels stecken, sodass noch etwa 7mm des Wolfram-Drahtes als Zwischenstück sichtbar ist (siehe Bild 1).

- In den bereitgestellten Glasbecher werden nun ca. 250ml der KOH-Lauge eingefüllt und in die Aussparung für den Becher der Spitzenätzstation gestellt (siehe Bild 2).
- Anschließend wird die Kathode (ringförmiger Draht) in den Becher gehängt und mittels Krokodilklemme mit dem Kathodenanschluss der Spitzenätzstation verbunden (siehe Bild 2-a).

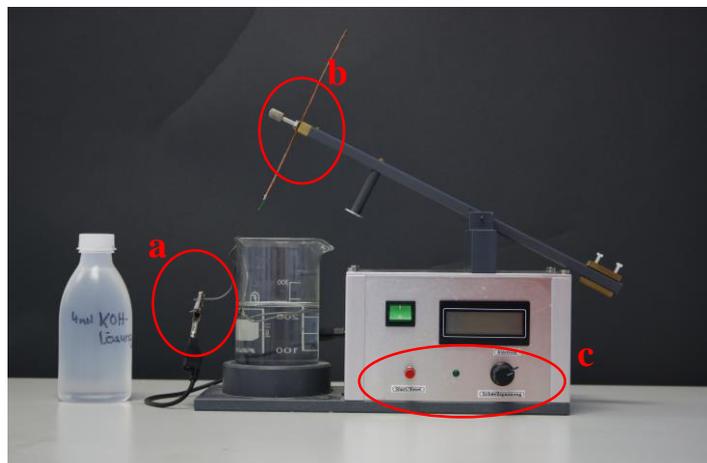


Bild 2: Spitzenätzstation

- Als nächstes wird der Kipparm in waagrechte Stellung gebracht und in dieser gehalten. Währenddessen wird der Kupferdraht soweit in der Einspannvorrichtung (siehe Bild 2-b) abgelassen bis sich der sichtbare Teil des Wolfram-Drahts, etwa zur Hälfte in der Flüssigkeit befindet (siehe Bild 3).
- Anschließend wird die Spitzenätzstation angeschaltet (grüner Schalter) und der Startknopf (roter Knopf) an der Spitzenätzstation gedrückt, dabei sollte auch beachtet werden, dass die Schwellspannung auf markierter Einstellung steht (siehe Bild 2-c)
- Nun beginnt der Ätzvorgang. Während des Vorgangs sollte peinlichst darauf geachtet werden, dass Bodenvibrationen durch Umherlaufen oder anstoßen des Tisches vermieden werden.
- Der Ätzvorgang ist beendet, sobald die mit Kabel eingefasste Spitze des Wolframdrahtes abfällt und der Kipparm in die Höhe schnell.
- Nun wird der Kupferdraht wieder aus der Einspannvorrichtung genommen und durch destilliertes Wasser gezogen. Dabei muss unbedingt darauf geachtet werden, dass die Spitze keinerlei Berührungen ausgesetzt wird.



Bild 3: Wolframdraht wird von Laugenoberfläche „halbiert“

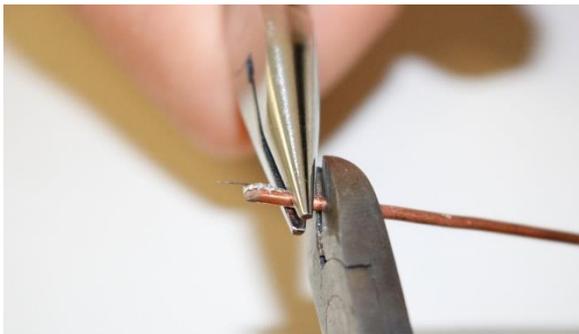


Bild 4: Zuschneiden der fertigen Messspitze

- Anschließend wird mit einer Spitzzange der Kupferdraht knapp über der Lötstelle gehalten und unmittelbar daneben, der Rest des Kupferdrahtes mit einem Seitenschneider abgezwickelt. (Als Richtwert sollte die Wolframspitze inklusive Kupferdraht nicht länger als 2 cm sein (siehe Bild 4).

Schritt 3: Messung vorbereiten

- Nun wird die Messspitze in die dafür vorgesehene Halterung des Piezos der Scaneinheit eingeschoben und mit der dort vorhandenen Schraube fixiert (Siehe Bild 5).
- Bevor die Scaneinheit wieder abgestellt wird, wird die zu untersuchende Probe unter den Probehalter, der gleichzeitig die Probenspannung anlegt, geschoben. Anschließend wird die Scaneinheit wieder so auf die Schwingungsdämpfungseinheit gestellt, dass die Scanspitze sich

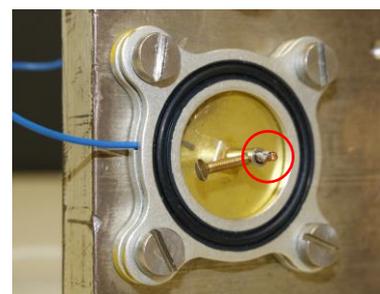


Bild 5: Eine in das Piezoelement eingespannte Messspitze (Unterseite Scaneinheit)



Bild 6: Fertig aufgestelltes STM mit DVD-Master als Probe vor der Annäherung(Nahansicht)

direkt über dem zu scannenden Abschnitt befindet, wobei darauf geachtet werden muss, dass keine der Füße (Schrauben) mit der Probe in Berührung kommt, da an dieser eine Spannung anliegt. Um die Vorspannung an die Probe anzulegen, wird der Bananenstecker der Scaneinheit in den Probehalter gesteckt (Siehe Bild 6).

- Nun werden beide Grobgewindeschrauben mit zwei Inbusschlüsseln parallel gegen den Uhrzeigersinn gedreht, bis die Scanspitze sich etwa einen Millimeter über der Probe befindet. Achtung das Ende der Spitze ist nur schwer sichtbar! (Siehe Bild 7)

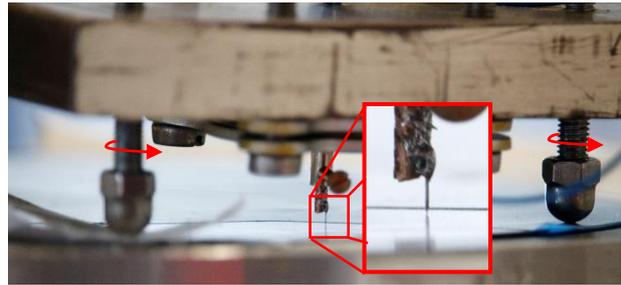


Bild 7: Pfeile: Drehrichtung zum nähern – Vergrößerung: fertig genäherte Messspitze

Während den nachfolgenden Schritten bis zum Ende des Scanvorgangs, muss absolute Stille herrschen und ein Umherlaufen oder Ähnliches absolut vermieden werden!

- Zur restlichen Annäherung wird der Schrittmotor verwendet. Hierzu öffnet man das STM-Messprogramm und drückt als erstes den Button „Open“ und anschließend „Annähern“. Nun wird solange gewartet, bis sich der Schrittmotor nichtmehr bewegt. Ist dies geschehen, so kann mit der Messung begonnen werden.

Schritt 4: Messung starten

- Vor Beginn der Messung kann unter Einstellungen der zu scannende Bildausschnitt gewählt werden. Dabei beschreibt x_{min} , x_{max} , y_{min} sowie y_{max} den jeweiligen Bildrand. Es bietet sich an, vorerst einen Scan mit maximalem Bildausschnitt zu starten.
- Anschließend kann im Hauptfenster noch die Schrittweite eingestellt werden. Hier empfiehlt es sich, mit 8er Schritten zu starten. Bei kleineren Bildausschnitten kann die Schrittweite verkleinert werden. Ist jedoch ein Scan des maximalen Bildausschnittes mit minimaler Schrittweite angedacht (Dauer ca. 5h), so wird empfohlen, vorher einen Probescan durchzuführen, um die Qualität der Spitze zu testen, da sonst unter Umständen der Scan umsonst wäre. Nun wird der Button „Scan starten“ gedrückt.

Schritt 5: Bild erstellen

- Ist die Messung abgeschlossen, wird die Meldung „Scan beendet“ ausgegeben. Nun kann mittels Klick „Bild berechnen“ der Scan visualisiert werden. Möchte man diese Grafik als interaktive 3D Grafik betrachten, so sollte über den Reiter „Datei“ → „Speichern“ das Bild gesichert werden.
- Nun wird das Programm ImageJ geöffnet. Dort kann mit Klick auf „File“ → „Open“ das soeben gescannte Bild geöffnet werden.
- Anschließend wird mittels dem Reiter „Plugins“ → „3D“ → „Interactive 3D Surface Plot“ das Bild als Interaktive 3D Grafik dargestellt. In dem neu geöffneten Fenster empfiehlt es sich, die Schieberegler auf ihre Wirkung zu testen, um eine optimale Darstellung zu erhalten.

8. Abschließende Gedanken

Es muss angemerkt werden, dass thematische Überschneidungen mit der Facharbeit, „Bau eines Rastertunnelmikroskops“ von Jonas Dittmann (2006) nicht zu vermeiden waren, da es ein Ziel für mich war, eine Arbeit zu erstellen, welche als eigenständige Erklärung des gesamten STM dient, ohne dass Fremdarbeiten zum Verständnis herangezogen werden müssen. Zur Vertiefung einzelner Kapitel (jeweils in den Kapiteln genannt) ist es dennoch empfehlenswert, die oben genannte Arbeit hinzuzuziehen.

Insgesamt kann das STM noch weiter ausgebaut werden. Es sollte ein Vorgehen entwickelt werden, welches sichert, dass kontinuierlich gute Scanresultate entstehen, was jedoch den Rahmen dieser Arbeit bei weitem überschritten hätte. Dennoch wurde eine sehr solide Basis entwickelt, bei der es letzten Endes nur noch um Feinheiten geht.

Das Programm könnte zudem noch leicht überarbeitet werden. Angedacht war eine 3D-Ansicht direkt in das STM-Messprogramm zu implementieren, was jedoch aufgrund von Zeitmangel nicht mehr möglich war.

Sehr wünschenswert wäre, erfolgreiche Untersuchungen von HOPG (Highly oriented pyrolytic graphite), eines Blu-ray-Masters oder Ähnlichem, bei welchem eine Vorhersage des Scannergebnisses möglich ist. Daher wäre eine Anschaffung weiterer Proben sinnvoll.

9. Literaturverzeichnis

- Ansgar, Meroth und Tolg, Boris. 2007.** Infotainmentsysteme im Kraftfahrzeug. s.l. : Springer-Verlag, 2007.
- Arduino. 2016.** Arduino. [Online] 18. September 2016.
<https://www.arduino.cc/en/Main/ArduinoBoardDue>.
- **2016.** Arduino Language Reference. [Online] 2016. [Zitat vom: 25. Juli 2016.]
<https://www.arduino.cc/en/Reference/HomePage>.
- Armbrust, Nico. 2016.** Uni Magdeburg. [Online] 3. Februar 2016. https://www.uni-marburg.de/fb13/researchgroups/of/introduction/stm.html?language_sync=1.
- Bai, Chunli. 2000.** Scanning Tunneling Microscopy and Its Application. s.l. : Springer Science & Business Media, 2000, S. 180, Abschnitt b).
- Böge, Wolfgang und Franke, Peter. 2013.** Arbeitshilfen und Formeln für das technische Studium: Elektrotechnik, Elektronik, Digitaltechnik, Steuerungstechnik. s.l. : Springer-Verlag, 2013, 2013.
- Chen, Yan. 2009.** Smart Cantilever Beams for Nanomanipulation. s.l. : ProQuest, 2009.
- Conrad Electronic SE . 2016.** [Online] 2016. [Zitat vom: 12. September 2016.]
<http://produktinfo.conrad.com/datenblaetter/950000-974999/967645-da-01-de-Schrittmotor.pdf>.
- Demtröder, Professor Dr. Wolfgang. 2005.** Experimentalphysik Band 3. *Experimentalphysik Band 3*. Kaiserslautern : Springer-Verlag, 2005, S. 123.
- Gehring, Andreas. 2003.** Simulation of Tunneling in Semiconductor Devices . [Online] November 2003. <http://www.iue.tuwien.ac.at/phd/gehring/node83.html>.
- Goodfellow GmbH. 2016.** Goodfellows. [Online] 02. September 2016. www.goodfellows.com.
- Hund, Dr. Michael. 2016.** *Messungsvorgang bei Cassy Lab 2 - Sensor*. [Befragte Person] Peter Sprenger. 15. September 2016.
- Jonas, Dittmann. 2006.** *Bau eines Rastertunnelmikroskops*. 27. Januar 2006. indirekt.
- Klinke, Melanie. 2014.** Rastertunnelmikroskopie und-spektroskopie an Ruthenaten und der violetten Lithium-Molybdän-Bronze. s.l. : kassel university press GmbH, 2014, S. 9-11.
- Löfken, Jan Oliver. 2016.** Wissenschaft Aktuell. [Online] 19. Juli 2016. http://www.wissenschaft-aktuell.de/artikel/Pro_Atom_ein_Bit_1771015590172.html.
- Microsoft Corporation. 2016.** Microsoft - Visual Basic .Net. [Online] 2016.
<https://msdn.microsoft.com/de-de/library/2x7h1hfk.aspx>.
- **2016.** Microsoft. [Online] 2016. [Zitat vom: 25. Juli 2016.] [https://msdn.microsoft.com/de-de/library/system.drawing.color\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/system.drawing.color(v=vs.110).aspx).
- MISUMI Europa GmbH. 2016.** Misumi Europa. [Online] 28. August 2016. <http://www.misumi-europe.com/de/e-catalog/vona2/mech/M1300000000/>.

Peratech Holdco Limited. 2016. Peratech. [Online] 26. August 2016.

<https://www.peratech.com/what-is-qtc.html>.

Pololu Corporation. 2001-2016. Pololu Robotics & Electronics. [Online] 2001-2016. [Zitat vom: 13. September 2016.] <https://www.pololu.com/file/0J450/A4988.pdf>.

Rehm, Prof. Dr.-Ing. W. 2016. TU Chemnitz, Ausarbeitung zum Proseminar IBM-PC (SS 1997).

[Online] 3. Februar 2016. https://www.tu-chemnitz.de/informatik/RA/news/stack/kompodium/vortraege_97/dvd/dvdtext.html.

Reuß, Sebastian. 2011. Nanotechnologie im Schulunterricht. [Online] 20. Januar 2011. [Zitat vom: 28. August 2016.] <http://thomas-wilhelm.net/arbeiten/Nanotechnologie.pdf>.

Sonnleitner, Tobias. 2003 . Universität Regensburg. *Das Tunnelelektronenmikroskop*. [Online] 11. Dezember 2003 . <http://www.physik.uni-regensburg.de/forschung/schwarz/Mikroskopie/10-STM.pdf>.

Texas Instruments Incorporated. 2016. Texas Instruments. [Online] 30. August 2016.

<http://www.ti.com/lit/gpn/tlc2202>.

—, **2016.** Texas Instruments. *Use of Rail to Rail Operational Amplifiers* . [Online] 30. August 2016.

<http://www.ti.com/lit/an/sloa039a/sloa039a.pdf>.

Universität Ulm. 2016. Universität Ulm Praktikum Physikalische Chemie. [Online] 03. Februar 2016.

http://www.uni-ulm.de/physchem-praktikum/media/fp/v_11.pdf.

10. Abbildungsquellen

- Alle Abbildungen, welche nicht eigenen Ursprungs sind, wurden als solche gekennzeichnet.
- Sämtliche Diagramme, wurden mit Microsoft Excel erstellt.
- Das Struktogramm wurde mithilfe des Programmes Struktogramm-Editor Kevin Krummenauer erstellt. (<http://www.whiledo.de/index.php?p=struktogrammeditor> Stand 28.08.2016)
- Sämtliche Software-Screenshots wurden mithilfe dem Windowseigenen Tools „Snipping Tool“ gefertigt.
- Alle Schaltpläne wurden mit der Software „Eagle“ von CadSoft erstellt.
- Alle übrigen Digitalfotografien wurden mithilfe einer Canon Eos 750D erstellt.

A Anhang

Sämtliche benutzte, bzw. entstandene Dokumente und Dateien befinden sich auf der Anhangs-CD. Ebenso sind dort sämtliche entstandenen Scanresultate zu finden.

B Erklärung nach §29 LPO-I

Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Alle Stellen, die wörtlich, annähernd wörtlich oder dem Sinn nach aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Des Weiteren erkläre ich, dass ich mich keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Diese Versicherung gilt auch für Abbildungen, Tabellen und ähnliches.

Die vorliegende Arbeit wurde in dieser oder ähnlicher Form an keiner weiteren Prüfungsstelle eingereicht.

Würzburg, den