

FACHARBEIT
aus dem Fach
PHYSIK

Thema: RCL – Ein über das Internet ferngesteuertes Laborexperiment

Verfasser: Johannes Götzfried
Leistungskurs: 3Ph1
Kursleiter: OStR Hans-Christoph Schulz
Abgabetermin: Fr., 30. Januar 2009

Erzielte Note: _____ in Worten: _____

Erzielte Punkte: _____ in Worten: _____

(einfache Wertung)

Abgabe im Sekretariat am _____

(Unterschrift des Kursleiters)

Inhaltsverzeichnis

1	Einleitung	3
2	Allgemeines zu RCLs	3
2.1	Vorteile	3
2.2	Anforderungen	4
3	Realisierung eines RCLs zur Potentialmessung	4
3.1	Überblick	4
3.2	Hardwareschicht	4
3.3	Erste Softwareschicht	6
3.4	Zweite Softwareschicht	7
3.5	Darstellungsschicht	8
3.6	Physikalischer Aspekt	11
4	Ausblick	12
5	Fazit	13
A	Beispielmessungen einiger Elektrodenformen	14
A.1	Homogenes Feld	14
A.2	Spitze Elektrode	15
A.3	Kreiselektrode	15
	Literatur	18

1 Einleitung

RCL steht für „*Remote Controlled Laboratory*“ und bezeichnet eine reale Versuchsanordnung, die über das Internet steuerbar ist. In der Regel besteht ein RCL aus einem geeigneten Experiment, einer Webcam, mit der dieses beobachtet werden kann, und einer Website, die eine möglichst einfache Steuerung mit nahezu jedem Webbrowser erlaubt (Abbildung 1).

Am Anfang dieser Arbeit wird auf allgemeine Vorteile und Anforderungen von RCLs eingegangen. Im weiteren Verlauf soll ein Beispiel-RCL zur Potentialmessung, das im Rahmen dieser Arbeit angefertigt wurde, vorgestellt und am Schluss ein kleiner Ausblick auf mögliche zukünftige Tätigkeiten gegeben werden.

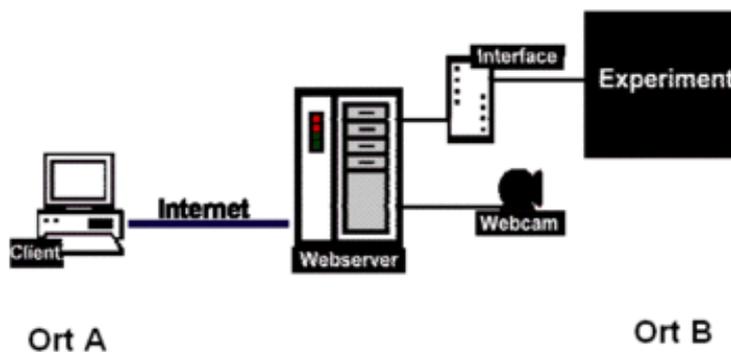


Abbildung 1: Schema eines RCLs

2 Allgemeines zu RCLs

2.1 Vorteile

Die Vorteile von RCLs gegenüber herkömmlichen Experimenten liegen auf der Hand. Benutzer eines RCLs sind völlig unabhängig von Zeit und Ort, da sie das Experiment zu jedem Zeitpunkt fernsteuern können und die einzige Voraussetzung ein funktionierender Internetzugang ist. Des weiteren können mehrere Personen den gleichen Aufbau nutzen, was vor allem bei sehr teuren oder gefährlichen Experimenten, wie zum Beispiel Versuchen mit radioaktiver Strahlung, von Vorteil ist. Auch in der Schule könnten RCLs sinnvoll eingesetzt werden, da der Aufbau eines lokalen Versuchs eingespart werden kann und Schüler das Experiment zu Hause wiederholen können. Um diese Vorteile zu gewährleisten, sind allerdings verschiedene Anforderungen an RCLs zu stellen.

2.2 Anforderungen

RCLs müssen wesentlich robuster und weniger fehleranfällig als Realexperimente sein. Es muss für jede mögliche Benutzereingabe ein genau definiertes Verhalten geben, damit die Funktionalität dauerhaft gewährleistet ist oder im Extremfall zumindest eine Beschädigung des Versuches verhindert wird. Damit man das Experiment gut verfolgen und damit nachvollziehen kann, ist es erforderlich, dass der Versuch nur so schnell abläuft, dass er bei einer gewöhnlichen Internetverbindung über eine Webcam zu beobachten ist. Schließlich ist es notwendig, eine Art Zugriffsmanagement zu implementieren, um zu verhindern, dass mehrere Leute gleichzeitig die Steuerung übernehmen und sich gegenseitig behindern. Diese Anforderungen waren unter anderem auch in folgendem Beispiel-RCL zu berücksichtigen.

3 Realisierung eines RCLs zur Potentialmessung

3.1 Überblick

Im Rahmen dieser Arbeit wurde ein vollständiges RCL angefertigt, dessen Funktionsweise hier ausführlich beschrieben wird. Das Experiment, eine Potentialmessung im elektrolytischen Trog, stellt ein klassisches physikalisches Problem dar. Dabei werden die Potentialverteilung und das dadurch festgelegte elektrische Feld, das von verschiedenen geladenen und geformten Elektroden im Wasserbad erzeugt wird, durch einen in zwei Raumrichtungen beweglichen Fühler gemessen und ausgewertet.

Die Steuerung des Experiments lässt sich grob in vier Schichten unterteilen, die aufeinander aufbauen. Beginnend mit der *Hardwareschicht*, die den Aufbau des Experiments und die elektronische Schaltung vereint, über die *Erste Softwareschicht*, die die Steuerung durch den Mikroprozessor beinhaltet, die *Zweite Softwareschicht*, die die Verwaltung durch den Computer verkörpert, bis zur *Darstellungsschicht*, die dem Benutzer eine ansprechende Oberfläche für die Steuerung bietet, soll die Realisierung des RCLs in diesem Abschnitt näher erklärt werden. Am Ende des Abschnitts soll noch einmal ausführlich auf den *physikalischen Aspekt* des Experiments eingegangen werden.

3.2 Hardwareschicht

Der bewegliche Fühler wird durch einen sogenannten „X-Y-Plotter“ ermöglicht, der mit Hilfe von zwei Schrittmotoren jeden Punkt einer Ebene ansteuern kann. Schrittmotoren

haben keinen Kommutator, d. h. die Umpolung erfolgt nicht automatisch, und somit muss die gezielte Ansteuerung der Spulen direkt von einem Mikroprozessor übernommen werden. Sie haben jedoch den Vorteil, dass das Getriebe eingespart und jede Position exakt angesteuert werden kann. Dies ist ein wichtiger Punkt, um die geforderte geringe Fehleranfälligkeit zu garantieren.

Als Mikroprozessor kommt hier der R8C/13 von Renesas zum Einsatz, der im nächsten Abschnitt bei der Darstellung der Programmierung noch genauer vorgestellt wird. Der R8C/13 ist über eine serielle Schnittstelle, also genauer gesagt mit „USB to Serial Converter“, mit dem Computer verbunden. Über diese USB-Verbindung kommt auch die Spannungsversorgung von +5V für den Mikroprozessor und die Elektroden zustande.

Neben dem Mikroprozessor und weiteren Bauteilen wie zum Beispiel Resettaster, Widerständen und Kondensatoren, deren Anordnung dem Schaltplan der Übungsplatine[6] zu dem Prozessor entnommen wurde, ist auf der Platine, die im Vordergrund von Abbildung 2 zu sehen ist, noch ein ULN2803 Treiberbaustein[7] verlötet. Dieser Treiberbaustein ist für die Ansteuerung der einzelnen Spulen in den Schrittmotoren zuständig. Er ist notwendig, da der Prozessor allein weder die richtige Spannung liefert noch genügend Stromstärke vertragen würde, um die Spulen direkt anzusteuern.

Die Funktionsweise des Bausteins ist folgendermaßen zu erklären: Jeder Eingang des ULN2803 ist mit einem Digitalausgang des Prozessors verbunden und jeder dem Eingang gegenüberliegende Ausgang des Treiberbausteins ist mit einem Teil der Motorspule verbunden, während der andere Teil der Spule auf +12V einer externen Spannungsversorgung liegt (Abbildung 2, Hintergrund). Wenn also der Mikroprozessor einen Digitalausgang auf „high“ (von 0V auf +5V) schaltet, dann schaltet der Treiberbaustein den Ausgang von +12V auf 0V. Damit liegt die entsprechende Spule des Motors zwischen 0V und +12V und kann somit ihr Magnetfeld aufbauen. Durch diese Schaltung ist eine softwaremäßige Ansteuerung jeder einzelnen Motorspule möglich, indem der entsprechende Digitalausgang im Programm aktiviert wird.

Der Fühler besteht aus einem dünnen Draht, der direkt an einen Analogeingang des Mikroprozessors führt. Die Elektroden sind wahlweise auf 0V oder auf +5V gelegt. Vor die +5V-Elektrode ist zum Schutz im Falle eines Kurzschlusses ein 10k Ω -Widerstand geschaltet. Der Prozessor kann somit am Analogeingang ein Potential zwischen 0V und +5V messen (genauer: zwischen 0V und der Eingangsspannung des Prozessors) und diesen Wert zurückgeben.

Abbildung 2 zeigt den vollständigen Versuchsaufbau. Nötig sind noch die Lampe, um den Versuch zu beleuchten, und die Webcam, die die Position des Fühlers im Wasserbad

für die Benutzer sichtbar macht. Durch diese Anordnung ist mit entsprechender Software, die im Folgenden vorgestellt wird, die Steuerung des Experiments möglich.



Abbildung 2: Experimentaufbau

3.3 Erste Softwareschicht

Diese Schicht bezieht sich auf die Software, die auf dem R8C/13 Mikroprozessor ausgeführt wird. Der Prozessor besitzt zwei serielle Schnittstellen für die Verbindung zum Computer, mehrere Digitalein- und -ausgänge, einen A/D-Wandler, der es ermöglicht, an jedem Digitaleingang auch ein analoges Signal zu empfangen, und weitere Features, wie zum Beispiel Timer und Interrupts, die aber in dieser Arbeit nicht benötigt werden. Vorteilhaft ist, dass man diesen Prozessor flashen (d. h. ein neues Programm aufspielen) kann, ohne dass dazu ein zusätzliches Programmiergerät erforderlich wäre.

Das Programm, das hier zum Einsatz kommt, ist bis auf die Startroutine, die in der Programmiersprache Assembler geschrieben ist, in C geschrieben. Es war der aufwändigste Teil der gesamten Arbeit, da der Umgang mit dem Prozessor erst erlernt und essentielle Funktionen zum Teil selbst implementiert werden mussten. Als Compiler kam der quelloffene GCC[8] zum Einsatz, der unter anderem die R8C Familie nativ unterstützt. Das komplexe Programm selbst hat einen recht einfachen Ablauf: Der Prozessor wartet, bis ein Kommando vom Computer ankommt, zerlegt es und führt eine Aktion durch oder meldet einen Fehler zurück. Unter den möglichen Kommandos sind für die Arbeit

nur zwei relevant. Das erste Kommando veranlasst den Mikroprozessor, den Analogwert an der aktuellen Position zu messen und zurückzugeben, mit dem zweiten ist es möglich, einen der beiden Motoren um eine bestimmte Anzahl an Zyklen in eine Richtung drehen zu lassen, wobei 25 Zyklen einer vollen Umdrehung entsprechen. Bei größerem Interesse zur Funktionsweise dieser Software wird empfohlen, den Quellcode zu lesen, der zusammen mit dieser Arbeit veröffentlicht wird.

3.4 Zweite Softwareschicht

Die zwei oben vorgestellten Kommandos werden in der zweiten Softwareschicht erzeugt und entsprechend vom Computer an den Prozessor gesendet. Dazu läuft auf dem PC, der zur Verwaltung des Experiments bereitsteht, unter einem Debian GNU/Linux¹ System ein sogenannter Dämon. Man bezeichnet einen Prozess als Dämon, wenn er vollständig im Hintergrund abläuft und nur noch indirekt mit dem Benutzer kommuniziert. Dieser Dämon, der schlicht „rclid“ benannt wurde, spielt eine entscheidende Rolle, indem er auf der einen Seite die Benutzerzugriffe kontrolliert, das heißt, er speichert, wer gerade eingeloggt ist und wie lange derjenige das Experiment noch steuern darf. Auf der anderen Seite kommuniziert er mit dem Prozessor und hält den aktuellen Status des Prozessors, also „beschäftigt“ oder „aufnahmefähig“, fest. Eine sehr wichtige Aufgabe des Dämons ist darüber hinaus die Speicherung der aktuellen Position, um zum Beispiel Grenzen setzen zu können oder die gemessenen Ergebnisse zur grafischen Darstellung an ein zusätzliches Programm weiterzugeben.

Da besonders bei Benutzerkommandos, die einen längeren Zeitraum zur Ausführung benötigen, häufig während noch mit dem Prozessor kommuniziert wird, weitere Benutzeranfragen akzeptiert werden müssen, ist rclid nicht ein einziger Prozess, sondern besteht nach dem Start aus einem Hauptprozess mit zwei Kindprozessen, von denen einer für Verbindungen von Benutzern und der andere für die Ansteuerung des Mikroprozessors zuständig ist. Die Synchronisation dieser unabhängig voneinander arbeitenden Prozesse wurde mit Hilfe von Unix-Sockets realisiert, auf die hier aber nicht näher eingegangen wird.

Zusammenfassend ist festzustellen: Ein Kindprozess empfängt die Benutzerkommandos und leitet sie an den Hauptprozess weiter, der diese auf Gültigkeit und Benutzerrechte prüft. Danach sendet der Hauptprozess dem Kindprozess entweder eine Fehlermeldung zurück, antwortet direkt oder sendet eine entsprechende Anfrage an den zweiten Kind-

¹<http://www.de.debian.org>

prozess, der wiederum den Prozessor ansteuert, und antwortet dann entsprechend. Tritt bei diesem Verlauf ein Fehler auf, so wird versucht, den Vorgang zu wiederholen, schlägt auch dies fehl oder tritt ein schwerwiegender Fehler auf, so werden sofort alle Prozesse terminiert, um dem Benutzer die Kontrolle zu entziehen. Diese Struktur und Vorgehensweise ist notwendig, um die Forderungen aus 2.2 wie Zugriffsmanagement und geringe Fehleranfälligkeit bestmöglich zu erfüllen.

Der Ablauf kann durchaus noch komplexer werden, indem zum Beispiel nach einer Messung an mehreren Stellen ein Graph generiert wird. In diesem Fall kommuniziert der Kindprozess, der die Prozessoransteuerung übernimmt, zusätzlich noch mit dem freien Funktionszeichenprogramm Gnuplot², dessen erzeugtes Bild später über die Darstellungsschicht betrachtet werden kann.

Allein mit diesem Dämon könnte ein Benutzer theoretisch das komplette Experiment steuern, da der Kindprozess, der die Verbindungen von Benutzern akzeptiert, wie jeder andere Dienst im Internet erreichbar ist, vorausgesetzt, dies wird nicht bewusst blockiert. Es gibt zum Beispiel Kommandos für das Ein- und Ausloggen, zum Zurückgeben von verschiedenen Werten, die der Dämon speichert, wie zum Beispiel aktuelle Position oder die noch verfügbare Zeit eines Benutzers, zum direkten Messen des Potentials und zum unmittelbaren Steuern der Motoren. Aber auch komplexere Kommandos, wie das Ausmessen eines bestimmten rechteckigen Bereiches, sind möglich. Normalerweise werden die Befehle jedoch nicht direkt vom Benutzer gesendet, sondern über die Darstellungsschicht, da dies wesentlich komfortabler ist. Wer sich jedoch für das direkte Senden interessiert, der sollte einen Blick auf den Quellcode des Dämons werfen.

3.5 Darstellungsschicht

In diesem Abschnitt soll nun die endgültige Oberfläche, mit der man das Experiment üblicherweise über das Internet steuert, vorgestellt und alle konkreten Möglichkeiten, die ein Benutzer des Beispiel-RCLs hat, gezeigt werden. Die Oberfläche ist eine einfache Website, die in PHP geschrieben wurde. Diese PHP-Seite führt selbst nur wenige Fehlerüberprüfungen durch und sendet nahezu alle Benutzereingaben direkt an den Dämon, der dann einen eventuellen Fehler erkennen kann. Der Einsatz einer Website hat den entscheidenden Vorteil, dass sie mit fast jedem Browser und unter fast jedem Betriebssystem betrachtet werden kann, und das Experiment damit einem großen Benutzerkreis zugänglich gemacht wird.

²<http://www.gnuplot.info>

Als Webserver kommt der etablierte Apache Web Server³ zum Einsatz, der in diesem Fall wie der Webcamserver⁴ auf demselben Computer wie der oben beschriebene Dämon läuft. Dies muss nicht zwangsläufig so sein, es wäre durchaus denkbar, den Webserver auf einem anderen Computer laufen zu lassen.

Die Funktionsweise der Website ist schnell erklärt. Zuerst baut sie eine Verbindung zum Dämon auf und schickt je nachdem, was der Anwender angeklickt hat, einen entsprechenden Befehl an den Dämon. Danach sendet sie selbst weitere Anfragen, um vom Dämon die aktuelle Position, die Restzeit des Benutzers und Weiteres zu erfahren. Anhand dieser Informationen und dem Ergebnis des Benutzerbefehls wird dann dynamisch eine Website mit aktuellen Informationen und eventuellen Fehlermeldungen generiert und dem Benutzer angezeigt. Die Webcamübertragung ist mit einem sehr kurzen Javascript gelöst, das den Webbrowser des Benutzers anweist, sich nach einem vorgegebenen Zeitintervall ein neues Bild vom Webcamserver zu holen. Dazu ist es notwendig, dass der Benutzer Zugriff auf diese beiden Dienste, den Web Server und den Webcamserver, hat. Ist dies nicht der Fall, was zum Beispiel hinter sogenannten Proxies vorkommen kann, ist das RCL von diesem Ort nicht direkt erreichbar.

In der folgenden Abbildung 3 sieht man einen Screenshot der Website, über die das Experiment gesteuert wird.

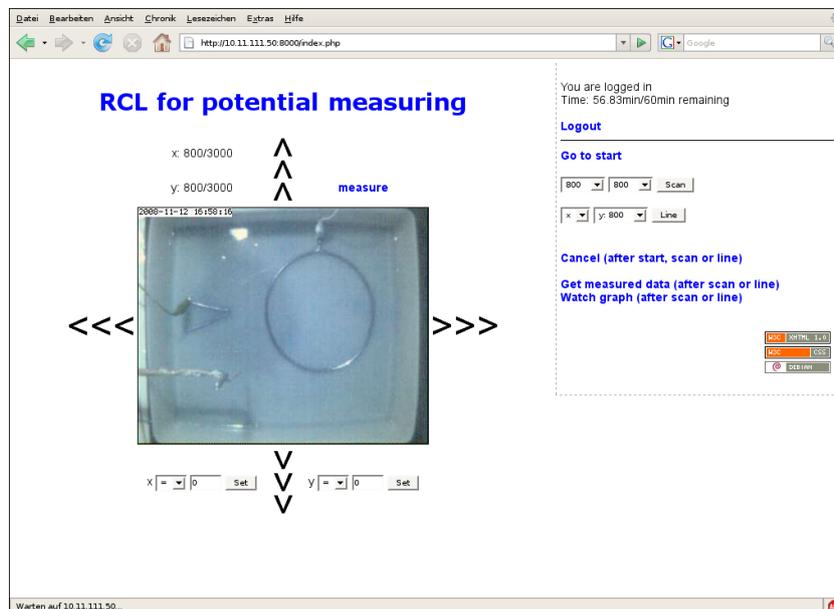


Abbildung 3: Screenshot der Website

³<http://httpd.apache.org>

⁴Debian-Paket: webcam-server

Das Erste, das dem Betrachter ins Auge fällt, ist das große Bild der Webcam, das ständig aktualisiert wird. Man sieht das Wasserbad mit den Elektroden und die ungefähre Position des Messfühlers. An allen vier Seiten des Bildes sind Pfeile angebracht, mit denen sich der Fühler direkt mit nur einem Mausklick steuern lässt. Je weiter der Pfeil vom Bild entfernt ist, desto weiter bewegt sich auch der Fühler. Wer eine exakte Fühlerposition ansteuern will, der kann einen genauen Wert für x und y in die Felder unter dem Webcambild eingeben, und der Messfühler fährt an diese Position. Über dem Bild sind links die aktuellen x - und y -Positionen und die jeweiligen Maximalwerte angegeben. Auf der rechten Seite befindet sich ein Link, um an der aktuellen Stelle den Potentialwert zu messen, der als Prozent der Eingangsspannung des Prozessors ausgegeben wird.

Am rechten Rand der Seite ist oben erkennbar, wer eingeloggt ist, und die Restzeit, die ihm für die Benutzung des Experiments bleibt. Es ist entweder ein Login- oder ein Logoutbutton vorhanden. Darunter findet sich eine einfache Funktion, die den Messfühler an den Start fährt. Die zwei interessantesten Funktionen des RCLs dürften die beiden folgenden sein: Mit der „Scan“-Funktion wird vom aktuellen Punkt, an dem man sich befindet, bis zu dem spezifizierten Endpunkt ein Feldscan durchgeführt, das heißt, der Fühler wird immer nur ein kleines Stück bewegt und dann wird jeweils gemessen, bis das gesamte Feld abgetastet ist. Bei der „Line“-Funktion wird auf einer bestimmten Achse in kurzen Abständen gemessen, so dass man zum Beispiel einen Querschnitt durch ein vorher bestimmtes Feld bekommt. In beiden Fällen kann man sich, wenn die Messung beendet ist, entweder den Graphen ansehen, der vom Dämon generiert wurde, oder die Datei mit den gemessenen Werten herunterladen. Mit dieser Datei kann man, wenn man auf seinem Computer zum Beispiel das in 3.4 erwähnte Gnuplot installiert hat, die Messergebnisse noch einmal optisch darstellen, beziehungsweise bestimmte Graphen erzeugen. Da diese beiden Funktionen teilweise sehr viel Zeit zur vollständigen Ausführung benötigen, kann man sie im Vergleich zu Positionsänderungen jederzeit abbrechen, erhält dann jedoch kein Messergebnis.

Alle x - und y -Werte sind auf dieser Website in den in 3.3 vorgestellten Zyklen angegeben. Falls irgendwelche falschen Eingaben gemacht werden oder man nicht eingeloggt ist, erscheint am Anfang der Seite eine Fehlermeldung, die in der Regel vom Dämon generiert wird. Für typische Elektrodenanordnungen werden am Ende dieser Arbeit noch einige auf die oben beschriebene Art erzeugte Beispielgraphen vorgestellt.

3.6 Physikalischer Aspekt

Nach der ausführlichen Erklärung zur Benutzung des RCLs und zur technischen Funktionsweise der Ansteuerung soll hier noch einmal explizit auf den physikalischen Aspekt der Potentialmessung im elektrolytischen Trog eingegangen werden.

Die verschieden geladenen Kupferelektroden erzeugen ein bestimmtes, elektrisches Feld, das durch die Potentialdifferenz zwischen dem jeweiligen Messpunkt und einem Vergleichspotential in einem schwach leitenden Elektrolyt – in den Versuchen dieser Arbeit normales Leitungswasser – näher bestimmt werden kann. Bei Experimenten mit Salzwasser hat sich gezeigt, dass Leitungswasser geeigneter ist, da anderenfalls eine Elektrolyse einsetzt. Ganz allgemein gilt zwischen der elektrischen Feldstärke \vec{E} und der Potentialdifferenz U folgende Beziehung[10]:

$$U_{12} = \int_1^2 \vec{E} \circ d\vec{s} \quad (1)$$

Hier ist U_{12} also die Potentialdifferenz von P_1 nach P_2 und \vec{s} der Weg von P_1 nach P_2 . Da sowohl der Weg \vec{s} als auch die elektrische Feldstärke \vec{E} Vektoren sind, wird somit das Skalarprodukt benötigt, und es folgt daraus außerdem, dass Linien gleichen Potentials beziehungsweise mit $U_{12} = 0$, sogenannte Äquipotentiallinien, auf den Feldlinien senkrecht stehen müssen. Mit diesem Hintergrundwissen fällt es später leicht, die Richtigkeit der Messungen zu überprüfen, indem die Potentialverteilung für bekannte Elektrodenanordnungen berechnet und mit dem Graphen der gemessenen Werte verglichen wird.

Das Vergleichspotential ist in diesem Fall 0V, das auch am Prozessor anliegt und über die USB-Schnittstelle vom Computer kommt. Normalerweise sollte man somit bei Elektroden, die auf 0V oder auf die Eingangsspannung des Prozessors (im Idealfall +5V) gelegt sind, Messwerte zwischen 0% und 100% erhalten. Da in der Praxis allerdings Spannung, zum Beispiel durch Kabelwiderstände, verloren geht, bekommt man in der Regel nur Messwerte zwischen 0% und 30%. Das Leitungswasser hat einen sehr hohen Widerstand, so dass nur ein ganz geringer Strom fließen kann. Insgesamt wird bei dem gesamten Experiment Gleichspannung verwendet. Dies ist nicht optimal, da Wasser polar ist und somit durch elektrolytische Abscheidungen ein Spannungsverlust auftreten kann. Da der Prozessor jedoch nur mit Gleichspannung arbeitet, musste dieser Nachteil in Kauf genommen werden.

Im Anhang werden einige Beispielmessungen, die mit diesem RCL gemacht wurden, vorgestellt und physikalisch erläutert.

4 Ausblick

Trotz der zuverlässigen Funktionsweise des Beispiel-RCLs gibt es noch mehrere Verbesserungsmöglichkeiten, deren Realisierung entweder zu kompliziert gewesen wäre oder den Umfang dieser Arbeit überschritten hätte. Einige davon sollen an dieser Stelle genannt werden.

Aus physikalischer Sicht wäre es wohl besser, statt Gleichspannung an den Elektroden eine Wechselspannung anzulegen, da damit elektrolytische Abscheidungen vermieden werden können. Außerdem würde eine höhere Spannung sicher hilfreich sein, um Verluste zu kompensieren und genauere Messungen zu ermöglichen. In beiden Fällen wäre jedoch eine komplizierte elektrotechnische Schaltung notwendig, die diese spezielle Spannung, die zwischen den Elektroden an bestimmten Stellen gemessen wird, wieder in eine Spannung umwandelt, die der Mikroprozessor auswerten kann.

Sinnvoll wäre es ebenfalls, wenn die Lampe, die den Versuch beleuchtet, von der Ferne aus- und angeschaltet werden könnte. Im Prinzip wäre das gar nicht so kompliziert. Man benötigt wieder einen ULN2803 Treiberbaustein, dessen Eingang mit einem Digitalausgang des Prozessors verbunden ist und dessen Ausgang dann mit einem Relais verbunden wird. Dieses könnte dann problemlos die Spannungsversorgung der Lampe unterbrechen. Allerdings wäre hier äußerste Vorsicht geboten, da man mit normaler Netzspannung von 230V arbeitet.

Auch beim Zugriffsmanagement gibt es durchaus noch einige Verbesserungsmöglichkeiten. So könnte zum Beispiel ein Timeout bei Inaktivität dafür sorgen, dass Benutzer, die vergessen haben, sich auszuloggen, nicht für die gesamte restliche Zeit das RCL blockieren. Des weiteren könnte es hilfreich sein, weitere Kriterien, wie zum Beispiel Browseridentifikationen oder Cookies, zu benutzen, um einen Benutzer eindeutig als eingeloggt zu identifizieren, damit nicht mehrere Benutzer zum Beispiel hinter dem gleichen Gateway wie eine eingeloggte Person behandelt werden und sich so behindern.

Vor allem an der selbst geschriebenen Software, also am Dämon und am Programm, das auf dem R8C/13 läuft, gibt es sicher noch Verbesserungsvorschläge. Speziell beim Dämon könnten zum Beispiel noch erweiterte Sicherheitsmaßnahmen, die die maximale Anzahl an Verbindungen betreffen, die Verwendung von Threads anstatt von Kindprozessen und eine Möglichkeit, Einstellungen bei jedem Start zu ändern, implementiert werden. Auf diese Themen soll hier jedoch nicht näher eingegangen werden.

Dieser Abschnitt macht deutlich, dass es für die verwendete Versuchsanordnung viele weitere und interessante Aufgabenstellungen gibt, dieses RCL weiterzuentwickeln.

5 Fazit

Diese Arbeit stellt unter Beweis, dass mit einem nicht unerheblichen Zeitaufwand die Realisierung eines RCLs prinzipiell möglich ist. Voraussetzungen dafür sind natürlich eine gewisse Basisausstattung an Werkzeug und ausreichende Programmierkenntnisse. Es sei noch angemerkt, dass das gesamte Beispiel-RCL zur Potentialmessung nahezu ausschließlich mit Hilfe von freier⁵ Software entwickelt wurde, was sicherlich nicht ohne die Vorarbeit dieser genialen Entwickler möglich gewesen wäre. In Zeiten, in denen auch in deutsche Schulen der Computer immer mehr Einzug hält, könnten RCLs also durchaus dazu beitragen, die Experimentvielfalt zu fördern und zu erweitern.

⁵<http://www.gnu.org/philosophy/free-sw.html>

A Beispielmessungen einiger Elektrodenformen

A.1 Homogenes Feld

Das homogene Feld eignet sich hervorragend für einen ersten Test des Versuchsaufbaus, denn man erkennt sofort, ob die Messungen in etwa richtig sind. Zudem ist der Aufbau denkbar einfach. Es werden nur zwei verschieden geladene gerade Elektroden parallel in einem beliebigen Abstand in das Wasserbad gelegt. Im homogenen Feld ist die elektrische Feldstärke \vec{E} konstant und aus (1) in 3.6 folgt somit für die Potentialdifferenz:

$$U_{12} = \vec{E} \circ \vec{s} \quad (2)$$

Bewegt man sich nun in Richtung des Feldes, also in diesem Fall in x -Richtung, so sollte die Steigung ebenfalls konstant sein, was man in Abbildung 4 im Rahmen der Messgenauigkeit erkennen kann. Die Linien am Grund sind die in 3.6 bereits erklärten Äquipotentiallinien, auf denen das elektrische Feld senkrecht steht und deren Parallelität ebenfalls auf ein homogenes Feld hinweist.

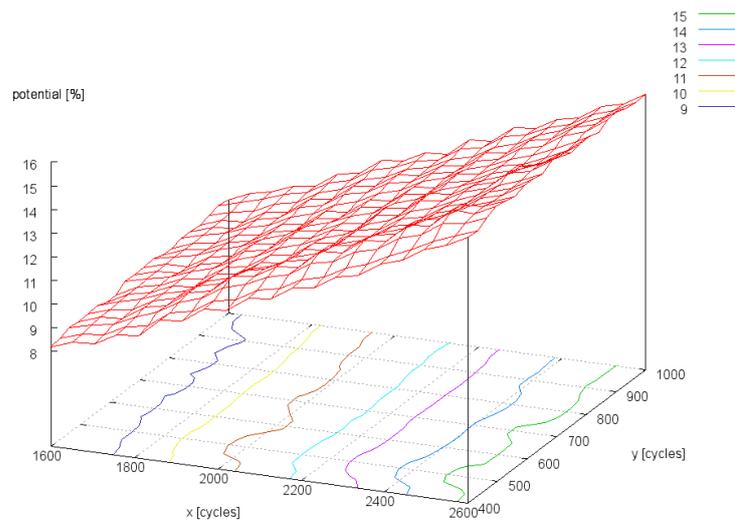


Abbildung 4: Graph des homogenen Feldes

A.2 Spitze Elektrode

Bei dieser Messung wurde eine spitze Elektrode und eine auf Bezugspotential gelegte gegenüberliegende gerade Elektrode verwendet. Bei einer solchen Anordnung ist nun die elektrische Feldstärke \vec{E} nicht mehr konstant und es können auch keine linearen Graphen oder parallelen Äquipotentiallinien mehr auftreten, was Abbildung 5 zeigt.

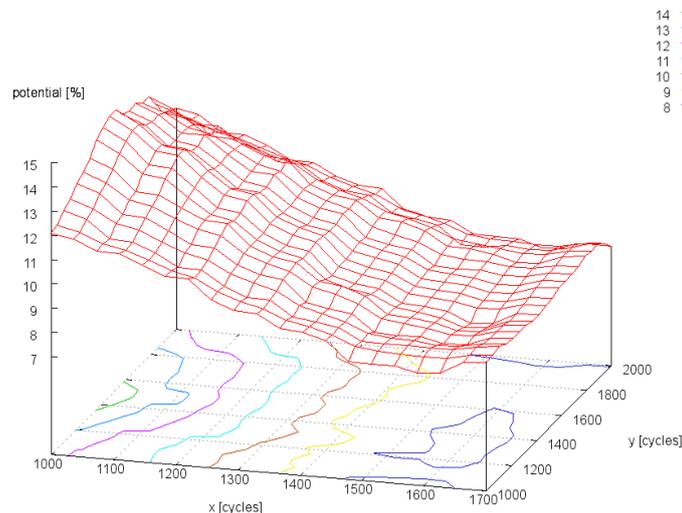


Abbildung 5: Graph einer spitzen Elektrode

A.3 Kreiselektrode

Als letztes Messergebnis soll hier noch eine Kreiselektrode vorgestellt werden. In diesem Fall sind beide Elektroden verschieden große Kreise, die denselben Mittelpunkt haben, wobei die größere Kreiselektrode auf Bezugspotential gelegt ist. Mit diesem Aufbau sollte man ein annähernd radialsymmetrisches Feld bekommen. Abbildung 6 zeigt den Graphen der Kreiselektrode und in Abbildung 7 sind nur die Äquipotentiallinien mit den genauen Koordinaten zu sehen. In einem radialsymmetrischen Feld gilt außerdem[10]:

$$\varphi = \frac{1}{4\pi\epsilon_0} \frac{Q}{r} = K \frac{1}{r} \quad (3)$$

Diese Beziehung gilt, wenn das Bezugspotential $\varphi_0 = 0$ im Unendlichen liegt und da die größere Kreiselektrode auf Bezugspotential liegt, kann dies angenommen werden. Mit der in 3.5 erläuterten „Line“-Funktion wird nun eine Messung auf der x -Achse durch

die Kreismittelpunkte gemacht. Da die Messung durch die Mittelpunkte gemacht wird, kann r durch x ersetzt werden, wenn vom Mittelpunkt in Richtung des Bezugspotentials gegangen wird. Wie in (3) zu sehen ist, sind alle Faktoren vor $\frac{1}{r}$ konstant. Um zu prüfen, ob die Messergebnisse diese Beziehung erfüllen, wird versucht, eine Funktion der Form $f(x) = a \frac{1}{x-b} + c$ zu finden, die zur rechten Seite des Graphen passt. Die Konstante K in obiger Beziehung (3) ist also hier a . Die Parameter b und c sind notwendig, da das Bezugspotential φ_0 weder im Unendlichen liegt, noch $\varphi_0 = 0$ gilt. In Abbildung 8 sind die Messergebnisse zu sehen, die auf einer Linie durch die Mittelpunkte liegen und die Funktion $f(x)$, die aus den Messergebnissen der rechten Hälfte berechnet wurde. Da diese Funktion im Vergleich zu anderen Funktionen, die ebenfalls getestet wurden, besonders gut zu den Messwerten passt, liefert sie zumindest einen Hinweis darauf, dass die Messergebnisse obiger Beziehung (3) in etwa genügen könnten. Es handelt sich also vermutlich um ein annähernd radialsymmetrisches Feld.

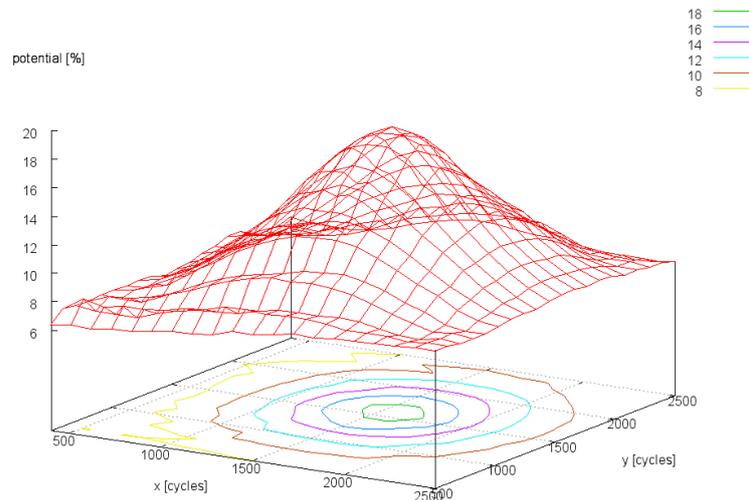


Abbildung 6: Graph einer Kreiselektrode

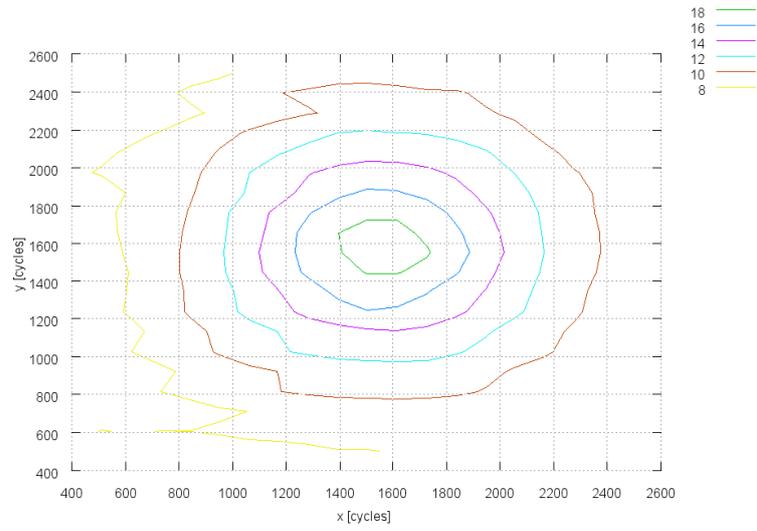


Abbildung 7: Äquipotentiallinien einer Kreiselektrode

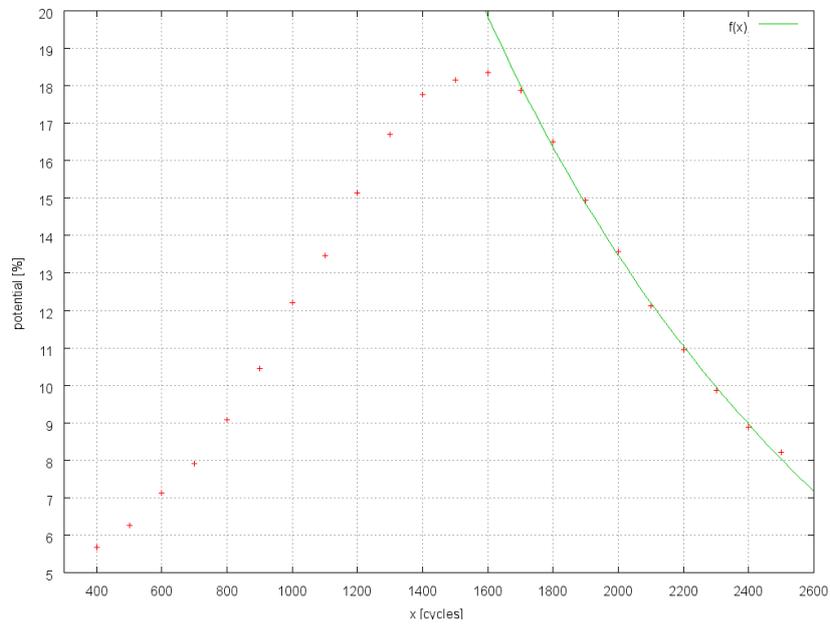


Abbildung 8: Messergebnisse auf einer Linie durch die Mittelpunkte

Literatur

- [1] RCL auf Wikipedia
http://de.wikipedia.org/wiki/Remote_Controlled_Laboratory
Stand: 2008-11-04

- [2] RCL auf dem Physik Bildungsserver Rheinland-Pfalz
<http://physik.bildung-rp.de/digitale-medien.html>
Stand: 2008-11-04

- [3] Jones on Stepping Motors [Tutorial über Schrittmotoren]
<http://www.cs.uiowa.edu/~jones/step/>
Stand: 2008-11-08

- [4] Renesas: R8C/13 Group – Hardware Manual, Rev. 1.20

- [5] Renesas: R8C/Tiny Series – Software Manual, Rev. 1.00

- [6] Schaltplan der Übungsplatine zum R8C/13
Elektor: Ausgabe 1/2006, Seite 25

- [7] Toshiba: ULN2803,04APG/AFWG Datasheet, Version: 2006-02-15

- [8] GCC für R8C/M16C/M32C
http://www.fischl.de/thomas/elektronik/r8c/r8c_gcc.html
Stand: 2008-11-10

- [9] Elektrolytischer Trog auf Wikipedia
http://de.wikipedia.org/wiki/Elektrolytischer_Trog
Stand: 2008-11-16

- [10] COULOMB-Potential und Elektrische Spannung U
Hammer/Hammer: Physikalische Formeln und Tabellen, 8. Auflage
Seite 41/42

Ich erkläre hiermit, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

_____, den _____
Ort Datum Unterschrift des Schülers