Hochschule Emden/Leer
Fachbereich Technik,
Abteilung Elektrotechnik und Informatik
Prof. Dr.-Ing. Dirk Rabe

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Dokumentation der Projektarbeit:

Entwicklung und Umsetzung einer Fernschreiberfunktion zwischen einem Raspberry Pi und dem historischen T 24-Fernschreiber der Firma Siemens & Halske AG.

von
Ralf Remmers
und
Raphael Boomgaarden

Betreuer:
Prof. Dr. Ing. Dirk Rabe,
Dipl.-Ing. Harald Buß

Erklärung zu Nutzungsrechten:
Wir, Ralf Remmers und Raphael Boomgaarden, übertragen die vollen Nutzungsrechte für alle eingereichten Unterlagen dieses Projektes an die Hochschule Emden/ Leer und deren Angehörigen. Dies gilt auch für die Weitergabe an Medien, die Veröffentlichung im Internet und die Nutzung im Rahmen öffentlicher Vorträge, Präsentationen und ähnlichem.

Eidesstattliche Erklärung:
Hiermit versichern wir, Ralf Remmers und Raphael Boomgaarden, dass wir die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Emden, den __.__.____

_____	_____
Ralf Remmers	Raphael Boomgaarden

Inhaltsverzeichnis

1 Einleitung ... 1
2 Das Gesamtsystem ... 2
3 Projektplanung und Verlauf ... 3
4 Der Fernschreiber ... 4
4.1 Stromlaufplan Motor ... 8
4.2 Stromlaufplan Signal ... 9
4.3 Modifikationen am Fernschreiber ... 10
5 Das Protokoll ... 15
5.1 Technische Daten ... 15
5.2 Die Übertragung ... 15
5.3 Die Codierung ... 17
6 Der Raspberry Pi ... 19
7 Die Software ... 21
7.1 Erläuterung der verschiedenen Modi ... 21
7.1.1 Modus 1 – Direkte Kommunikation ... 21
7.1.2 Modus 2 – Betrieb als Webserver ... 24
7.1.3 Modus 3 – Client für den Webserver ... 25
7.1.4 Befehlscodes ... 26
7.2 Ausgewählte verwendete Bibliotheken ... 27
7.3 Einrichtung der seriellen Schnittstelle auf dem Raspberry Pi ... 28
7.4 Umwandlung von ASCII in Baudot und zurück ... 29
7.5 Webinterface und Hotspot ... 32
7.6 Display ... 34
7.7 Drucker ... 35
8 Die Hardware ... 38
8.1 Schaltungen auf der Aufsteckplatine ... 38
8.2 Layout ... 40
8.3 Relais ... 43
9 Das Gehäuse ... 45
10 Messungen ... 49
11 Noch bestehende Fehler, bekannte Bugs ... 53
12 Fazit ... 54
Anhang ... 55
A Anleitung für die Inbetriebnahme ... 55
A.1 Anschlüsse ... 55
A.2 Modus 1 ... 57
A.3 Modus 2 ... 58
A.4 PHP-Seiten ... 59
A.5 Modus 3 ... 59
A.6 Programm Beenden ... 61

B Anleitung Vorführplatine 62
B.1 Aufbau 62
B.2 1. Teilstufe 63
B.3 2. Teilstufe 65
B.4 3. Teilstufe 66
C Anleitung zur Neuinstallation der Micro-SD-Karten 67
C.1 Mit Recovery-Images 67
C.2 Ohne Recovery-Images 67
D Weitere Bilder 69
E Abbildungsverzeichnis 70
F Tabellenverzeichnis 71
G Codeabschnittverzeichnis 71
H Literaturverzeichnis 72

1 Einleitung

Im Sommersemester 2016 fand die Veranstaltung „Praktikum Digitaltechnik“ statt, im Zuge dessen Herr Rabe und Herr Buß einen antiken Fernschreiber demonstrierten. Im Verlauf der Demonstration äußerten beide die Idee, mit dem Fernschreiber richtig kommunizieren zu können und nicht nur zu zeigen, dass die serielle Verbindung low-aktiv ist (Stromfluss ist eine Null). Daraufhin wurde unser Interesse geweckt und diese Projektarbeit ist entstanden, worin wir das Vorhaben erfolgreich umsetzten.
Besondere Herausforderungen waren, dass der Fernschreiber zwar ein Protokoll ähnlich dem RS-232 Protokoll verwendet, aber die Codierung der Zeichen nur fast einem sehr alten Standard entspricht und dass der Fernschreiber mit einer Stromschleife arbeitet. Zusätzliche Schwierigkeiten bot die Tatsache, dass der Fernschreiber nur einen Kanal für Dateneingang und -ausgang hat.

In dieser Dokumentation werden wir in Abschnitt 2 zunächst den Fernschreiber vorstellen. Danach wird das verwendete Protokoll beschrieben und wie dieses die Zeichen codiert.
Die Kommunikationsgegenstelle zum Fernschreiber, den Raspberry Pi 3, beschreiben wir in Abschnitt 4. Die von uns erstellte Software wird im darauffolgenden Abschnitt vorgestellt. Die Hardware, welche unter anderem die Signale vom Fernschreiber auf die vom Raspberry Pi umsetzt und umgekehrt, wird danach vorgestellt. Einige erfolgte Messungen werden ebenfalls vorgestellt, sowie die Konstruktion und der Bau eines Gehäuses. Zuletzt folgt noch eine Übersicht noch verbleibender Fehler. Im umfangreichen Anhang befindet sich eine Anleitung für die Inbetriebnahme, sowie eine Anleitung für eine Vorführplatine, welche dazu dient die Signale bequem auf einem Oszilloskop darzustellen. Sollte einmal eine der Micro-SD-Karten verloren gehen findet sich im Anhang eine Anleitung zur Neuinstallation der Micro-SD-Karten. Weiterhin befinden sich im Anhang noch Bilder.

2 Das Gesamtsystem

In diesem Abschnitt wird eine grobe Übersicht über den gesamten Aufbau des Projekts gegeben. Die einzelnen Teile werden dann in den nachfolgenden Abschnitten genauer beschrieben.

Der Aufbau des Projekts erfordert die von uns gebaute Aluminiumbox, den Fernschreiber, eine handelsübliche USB-Tastatur, zwei Laborleitungen und einen Trenntransformator. Fertig aufgebaut sieht das Ganze dann wie in Abbildung 1 aus. Näheres dazu in Abschnitt A „Anleitung für die Inbetriebnahme“ im Anhang.

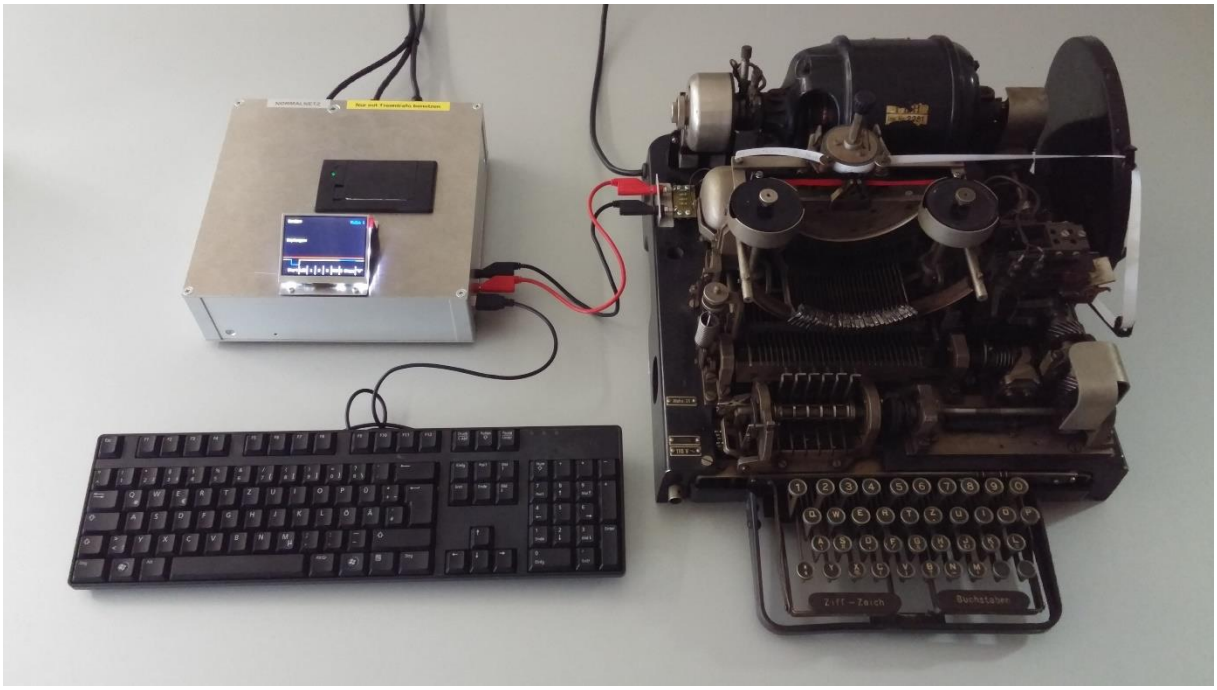

Abbildung 1 Projektaufbau

Hauptteil des Projekts ist die links im Bild zu sehende Aluminiumbox, in welcher sich ein Einplatinencomputer namens Raspberry Pi befindet (siehe Abschnitt 6). Auf diesem sitzt die von uns entwickelte Aufsteckplatine (siehe Abschnitt 8). Auch das Display wird von dem Raspberry Pi angesteuert (siehe Abschnitt 7.6).

Weitere Komponenten, die sich ebenfalls im Gehäuse befinden, werden in Abschnitt 9 beschrieben.

Zusätzlich wurden auch Änderungen am Fernschreiber vorgenommen. Diese und eine genauere Beschreibung des Fernschreibers befinden sich in Abschnitt 4.

3 Projektplanung und Verlauf

Vor Beginn des Projekts wurden Meilensteine erarbeitet und die Vorgehensweise geplant. Die Meilensteine waren:

1. Genaue vom Fernschreiber benutzte Spannungspegel und Ströme herausfinden und dokumentieren, dabei die Codierung überprüfen, genauen Typ des Fernschreibers feststellen und Dokumentation suchen (RPi bestellen) -> ca. 1 Woche
2. Software entwickeln und dokumentieren, sowie testen -> ca. 2 Wochen
    a. Umwandlung ASCII -> Baudot-Code und zurück
    b. Kommunikation über serielle Schnittstelle (zunächst zwischen 2 RPI)
    c. Feinheiten (vielleicht PC -> RPI -> Fernschreiber)
3. Display in Betrieb nehmen -> ca. ½ Woche
4. Schaltung(en) entwickeln um die richtigen Ströme bereit zu stellen und zu schalten -> ca. 1 Woche
5. Eventuelle Erweiterungen: Gehäuse

Die meisten Meilensteine wurden in der vorgesehenen Zeit abgearbeitet, manche sogar früher. Was am Ende viel Zeit gekostet hat, waren die Erweiterungen, die wir noch zum Projekt hinzugefügt haben, z.B. der Drucker, das Gehäuse und noch weitere Dinge. Auch Änderungen am Fernschreiber haben ihre Zeit gekostet.

Für die Zukunft haben wir gemerkt, dass Meilensteine genauer definiert werden sollten und man genauer auf die Meilensteine achten sollte, um „Feature Creep“, also das Hinzufügen von Funktionen über die eigentlich Vorhergesehenen hinaus zu verhindern. Auch das Testen des Gesamtsystems mit allen Funktionen und Fehlerbehebung taucht bei unseren Meilensteinen nicht auf, was für zukünftige Projekte aber definitiv beachtet werden sollte.

4 Der Fernschreiber

Ein Fernschreiber ist im Grunde eine Schreibmaschine, die befähigt wurde seriell zu kommunizieren. Was so einfach klingt und heute mit ein wenig Elektronik zu lösen ist, war zu der Zeit wo die ersten Fernschreiber aufkamen eine schwierig zu bewältigende elektromechanische Aufgabe, was man schon allein an den Maßen des Fernschreibers sehen kann[1].

Der im Rahmen dieser Projektarbeit verwendete Fernschreiber ist so alt, dass keine Dokumentation zu diesem Gerät gefunden werden konnte. Die einzigen Bezeichnungen, die gefunden werden konnten, abgebildet in Abbildung 3, waren „Alpha. 29“ an der linken Seite mit einer Nummer darunter, die „70200“ ist. Der Motor auf der hinteren Seite ist von Siemens-Schuckert mit einer nominellen Leistung von 38W, das Typenschild ist in Abbildung 4 zu sehen.

Abbildung 2 Der verwendete Fernschreiber

An der vorderen Seite des Fernschreibers, dargestellt in Abbildung 2 sieht man die Tastatur und die Schreibmechanik. Auf der rechten Seite befinden sich 2 Relais, welche vermutlich das Senden und Empfangen steuern. Hinten rechts befindet sich der Papierrollenhalter, wo der Papierstreifen auf einer Rolle befestigt wird, da der Fernschreiber ein Streifenschreiber und kein Blattschreiber ist.

Angetrieben wird die Schreibmechanik von dem bereits erwähnten Motor auf der hinteren Seite, der auf der einen Seite der Achse ein Zahnrad zum Antreiben der Mechanik und auf der anderen Seite einen Drehzahlregler hat. Dieser Drehzahlregler ist ein Fliehkraftregler und basiert auf dem Prinzip einer Fliehkraftkupplung. Unterhalb des Drehzahlreglers sieht man die Anschlüsse für die

[1] Zum Vergleich: Der Fernschreiber ist 41,5 cm x 57 cm x 25 cm (B x L x H) groß, der Raspberry Pi, mit Erweiterungsplatine und Display ist etwa 8,8 cm x 6,5 cm x 5 cm groß.

Stromschleife und dahinter zwei Kondensatoren. Nicht auf dem Bild zu sehen ist ein Vorwiderstand auf der Rückseite.
Intensive Recherche und auch ein erster Anruf bei Siemens im Sommer 2016 haben keine näheren Informationen zu diesem Fernschreiber zutage gebracht, daher mussten nahezu alle relevanten Daten experimentell bestimmt werden. Anfang 2017 schloss Siemens den Umzug ihres Firmenarchives ab. Im Januar 2017 starteten wir daraufhin eine erneute Anfrage.
Es ergab sich, dass der hier beschriebene Fernschreiber ein Modell namens T 24 ist, Alpha 29 bezeichnet hierbei die Tastatur und nicht den Fernschreiber selbst, wie wir zunächst dachten. Hersteller war die Firma Siemens & Halske AG. Siemens & Halske AG war die nachrichtentechnischen Sparte von Siemens. Der Motor wurde von Siemens-Schuckertwerke AG bezogen, der starkstromtechnischen Sparte. Scheinbar stammt das Modell aus dem Jahre 1931. Somit ist dieser Fernschreiber ein sehr frühes Modell. Unser Dank geht hierbei an Herrn Dr. Frank Wittendorfer, Leiter des Siemens-Archivs, der für uns recherchiert hat. [1]
Sollten noch weitere Informationen zu dem Fernschreiber zu Tage gefördert werden, werden diese auf der Webseite zur Projektarbeit veröffentlicht.

Abbildung 3 Die einzige gefundene Bezeichnung am eigentlichen Fernschreiber

Abbildung 4 Typenschild des Motors

Was unsere Recherchen allerdings hervor brachten, war ein Fernschreiber namens „Alpha 32“, der ab 1932 hergestellt wurde, allerdings ist das Video[2] auch darüber die einzige Information, die wir finden konnten.

Im Verlaufe des Projekts wurde für den Fernschreiber neues Papier bestellt, welches eigens angefertigt wurde und ein neues Farbband, welches dank DIN-Normen immer noch verfügbar ist.

[2] siehe: https://www.youtube.com/watch?v=etVWKD5FEXo

Um zu überprüfen, ob ein Ersatz des Vorwiderstandes durch einen Blindwiderstand, realisiert durch einen Kondensator, möglich ist, wurde ein Stromlaufplan erstellt, zusätzlich wurde noch ein Stromlaufplan von dem Signalpfad erstellt.

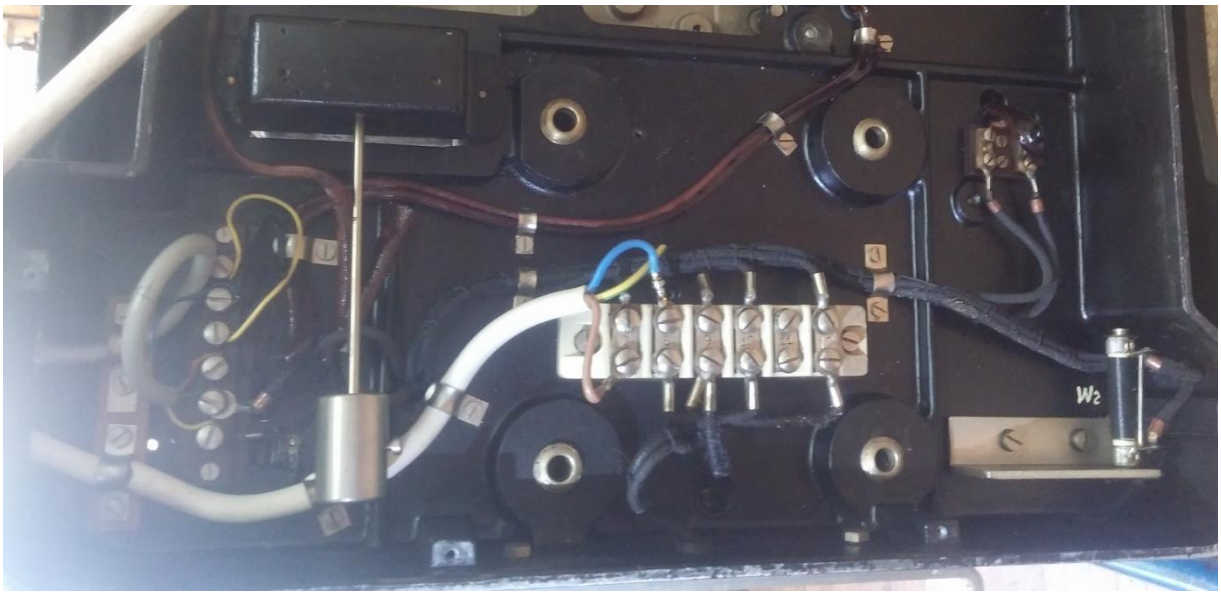

Abbildung 5 Verdrahtung des Fernschreibers

In Abbildung 5 ist die Verdrahtung des Fernschreibers zu sehen, wie sie vorgefunden wurde. Die Schraubleiste unten ist zur Verdrahtung des Motors, die Schraubleiste links für den Signalpfad. Die Nummern in den Stromlaufplänen beziehen sich auf diese Schraubleisten.
Um die Stromlaufpläne zu erstellen, wurden alle Leitungen von den Schraubleisten entfernt und einzeln mithilfe eines Ohmmeters nachverfolgt.

4.1 Stromlaufplan Motor

Im Endeffekt ergab sich der in Abbildung 6 vereinfacht dargestellte Stromlaufplan des Motors.

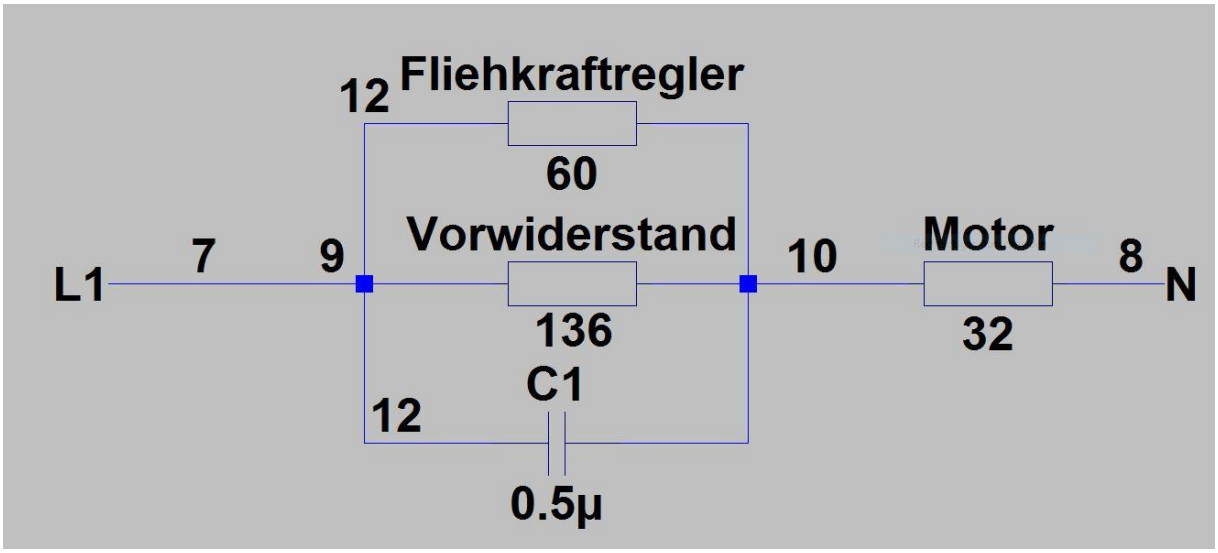

Abbildung 6 Stromlaufplan des Motors

Vereinfacht deshalb, weil es noch einen Pfad gab, der einfach tot gelegt war und in Reihe zu dem Kondensator war noch eine gebrückte Spule, wobei wir die Induktivität der Spule nicht gemessen haben, es könnte auch ein gewickelter Widerstand sein.
Mithilfe des Stromlaufplans kann man erkennen, wie die Drehzahlregelung des Motors funktioniert: Im Grundzustand ist der Fliehkraftregler geschlossen. Dadurch ist der gesamte wirksame Widerstand $60\,\Omega \parallel 136\,\Omega \parallel \frac{1}{2*\pi*50Hz*0{,}5\mu F} + 32\Omega = 73{,}36\,\Omega$. Der Strom ist also $\frac{230\,V}{73{,}36\,\Omega} = 3{,}14\,A$. Wenn der Motor sich zu schnell dreht, wobei die Geschwindigkeit durch 2 Rädchen einstellbar ist, trennt der Fliehkraftregler. Dadurch ist nun ein Widerstand von $136\,\Omega \parallel \frac{1}{2*\pi*50Hz*0{,}5\mu F} + 32\,\Omega = 165{,}2\,\Omega$ wirksam. $\frac{230\,V}{165{,}2\,\Omega} \approx 1{,}39\,A$ fließen dann durch den Motor.
Durch den geringeren Strom fällt an dem Motor auch weniger Spannung ab, wodurch dieser wieder langsamer wird. Wenn er zu langsam wird, schließt der Fliehkraftregler wieder und mehr Strom fließt durch den Motor. Dies geschieht sehr oft, so dass der Motor in etwa eine konstante Drehzahl hält. Die Drehzahl ist durch zwei Drehrädchen einstellbar. Diese konstante Drehzahl ist wichtig, da die Baudrate von der Motordrehzahl abhängt. Der Kondensator dient vermutlich zur Entstörung und um die Spannung etwas zu glätten.

Bei Messungen im Betrieb schwankte der Strom etwa zwischen 1,4 A und 1,8 A mit einem gemessenen Mittelwert von 1,54 A kurz nach Inbetriebnahme. Mit zunehmender Betriebsdauer sinkt der fließende Strom, bis er sich bei ca. 1,33 A einpendelt.
Der geringe Strom bei längerem Betrieb ist damit zu erklären, dass der Motor zum Anlaufen ein hohes Drehmoment braucht, was bei einem Universalmotor zu einem hohen Strom führt.
Beim Betrieb wird immer weniger Drehmoment benötigt, wodurch der Strom wieder sinkt.

Da der Vorwiderstand parallel zu dem Fliehkraftregler liegt, war es nicht möglich ihn durch eine Kapazität zu ersetzen, da es bei einer Kapazität nicht möglich ist, die über sie anliegende Spannung sprunghaft zu ändern. Bei kleinen Kapazitäten hat dies einen glättenden Effekt, aber je größer die Kapazität, desto mehr Strom muss fließen um den Spannungsunterschied auszugleichen. Dies würde bedeuten, dass mehr Strom durch den Fliehkraftregler fließt als vorgesehen, was diesen über kurz oder lang beschädigen oder sogar zerstören könnte.

4.2 Stromlaufplan Signal

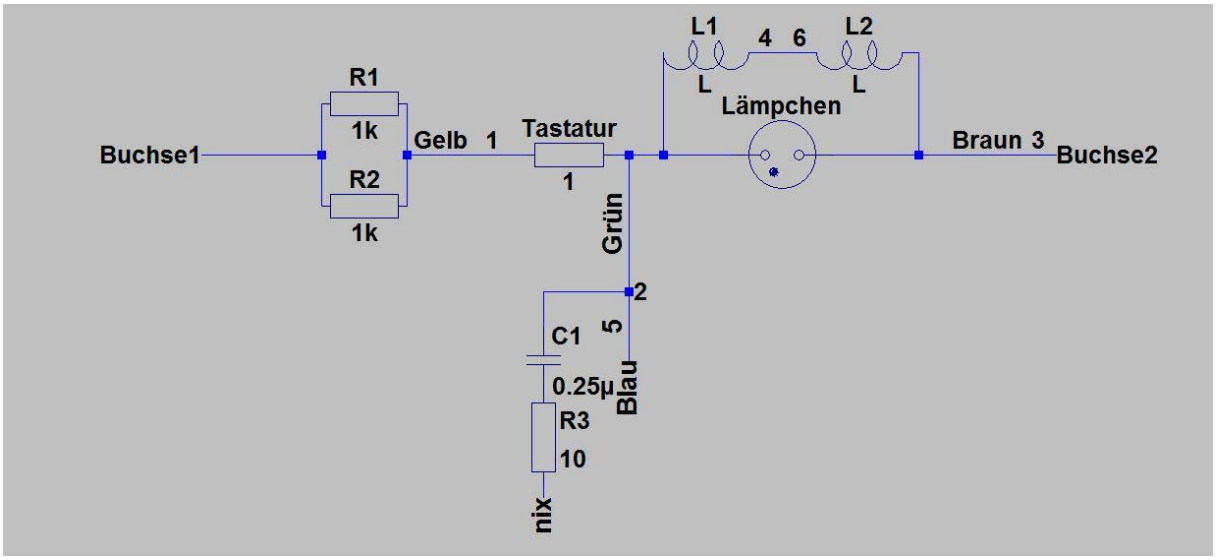

Abbildung 7 Stromlaufplan des Signalpfades

Die Verdrahtung des Signalpfads ist in Abbildung 7 erkennbar. Die Zahlen an den Leiterbahnen bezeichnen die Anschlüsse unterhalb des Fernschreibers und die Farben bezeichnen die Farben der vorgefundenen Leiter. Mittlerweile wurden einige davon getauscht, dazu später mehr.
Die beiden Widerstände R1 und R2 wurden nachträglich hinzugefügt, vermutlich um eine Strombegrenzung herbei zu führen. Es sind vermutlich zwei 1 W-Widerstände gewählt worden, da $P = I^2 * R = (40\,mA)^2 * 500\,\Omega = 0{,}8\,W$ sind und man bei 2 parallelen Widerständen eine Stromaufteilung bewirkt, wodurch nur 0,4 W pro Widerstand anfallen, was diese längerfristig haltbarer macht.
Die zwei Widerstände wurden später von uns durch vier 2 kΩ große 500 mW Widerstände ersetzt, dort fließen dann 10 mA pro Widerstand, somit ergibt sich die Leistung zu $P = I^2 * R = (10\,mA)^2 * 2\,k\Omega = 0{,}2\,W$ pro Widerstand. Dazu später mehr.
Das Widerstandssymbol mit dem Namen „Tastatur“ ist nur ein Platzhalter, da diese Leitung zur Tastatur geht und von dort wiederkommt, was dort passiert ist uns nicht ersichtlich. Von der Tastatur aus geht eine Leitung ab zu einem Kondensator, der in Reihe zu einem Widerstand geschaltet ist. Es könnte einmal die Funktion eines Filters gehabt haben, allerdings ist von uns der zweite Anschluss des Widerstands als nicht angeschlossen vorgefunden worden. Von dem Anschluss 2 aus war eine Brücke zu Anschluss 5, welcher auf der anderen Seite einen blauen Leiter angeschlossen hatte, der aber nirgends hinführte.
Von der Tastatur aus geht ein anderer Leiter auf einen Anschluss, der aussieht, als würde dort ein Lämpchen hingehören, allerdings befindet sich dort keines mehr.
Das Lämpchen ist überbrückt von 2 Relais und von dem Anschluss 3 geht es dann wieder zu der zweiten Anschlussbuchse.

4.3 Modifikationen am Fernschreiber

Am Fernschreiber wurden auch einige Veränderungen vorgenommen. Unter anderem wurden die 2 Kondensatoren an der rückwärtigen Seite gereinigt:

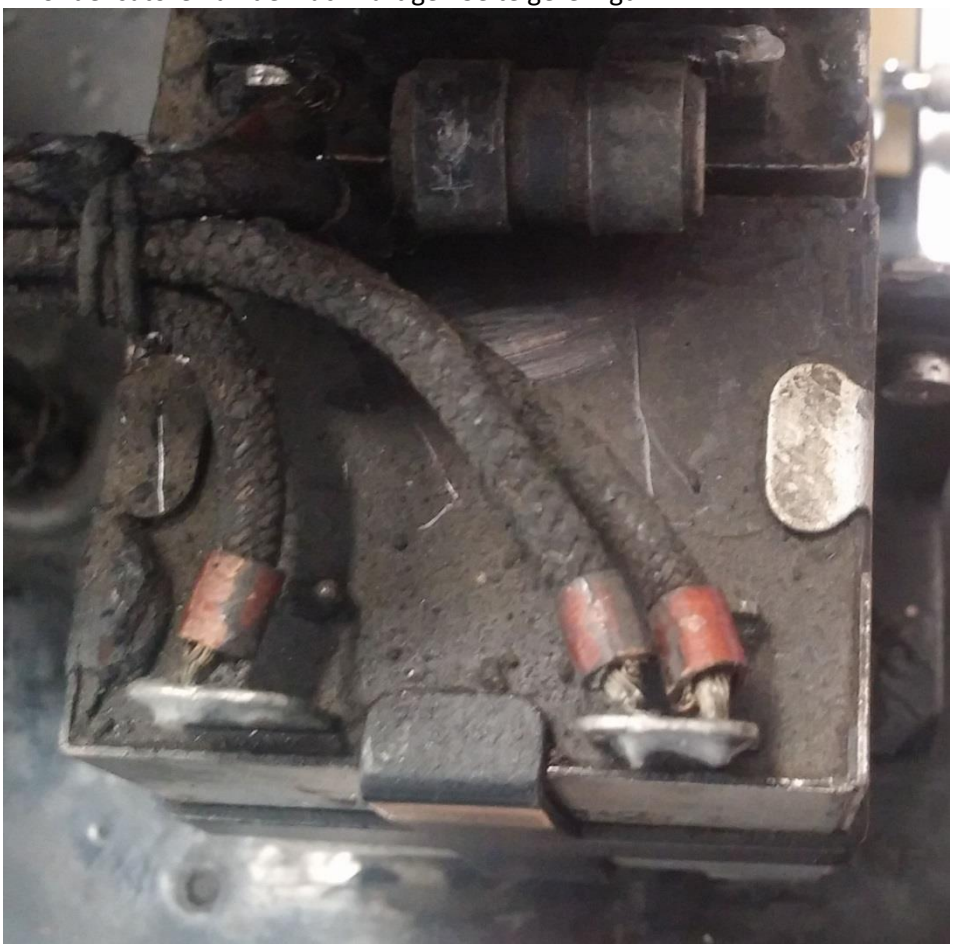

Abbildung 8 Kondensatoren von oben

In Abbildung 8 sind die beiden Kondensatoren von oben gezeigt, nachdem sie gereinigt wurden. Unten ein 0,5 µF Kondensator, oben ein 0,25 µF Kondensator. Nicht zu sehen sind weitere Aufschriften, wonach der 0,5 µF Kondensator auf eine deutlich über der Netzspannung liegende Spannung getestet wurde. Diese Aufschrift ist bei dem anderen Kondensator nicht vorhanden.
Die Beschriftung ist auf Abbildung 9 besser zu erkennen, dort ist links der 0,5 µF Kondensator und rechts der andere dargestellt. Man erkennt auch noch Reste des klebrigen Materials, welches bei der Reinigung entfernt wurde. Oberhalb des 0,25 µF Kondensators erkennt man einen runden Widerstand, der wohl mal zusammen mit dem Kondensator einen Filter gebildet hat, allerdings ist nur die eine Seite des Kondensators angeschlossen.

Abbildung 9 Kondensatoren von der Seite

Durch die Reinigung veränderte sich das Signal deutlich zum Positiven und wies danach wesentlich weniger Rauschen auf.

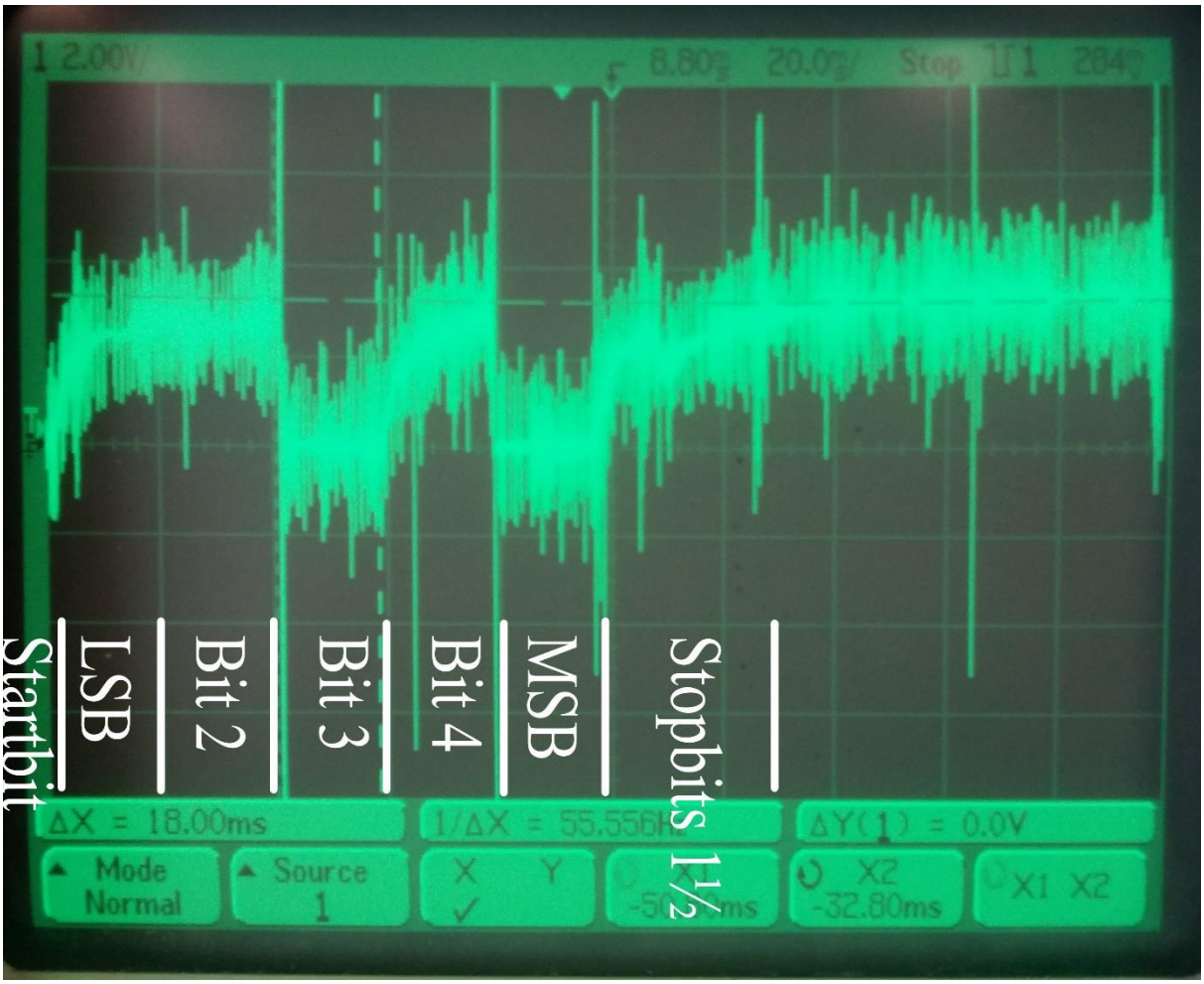

Abbildung 10 Signal vor der Reinigung, ein S wurde übertragen

Abbildung 10 zeigt das Signal vor der Reinigung, man erkennt deutlich das starke Rauschen. Gemessen wurde das Signal über einen Widerstand, da es ein Stromsignal ist. Nur der Fernschreiber war an eine Spannungsquelle angeschlossen.
Selbst ohne Filter war das Signal deutlich besser nach der Reinigung, man erkennt deutlich weniger Rauschen, wie in Abbildung 11 zu sehen. Woran genau dies liegt, ist nicht abschließend zu klären, unsere Vermutung ist, dass wir einen Kurzschluss bereinigt haben.

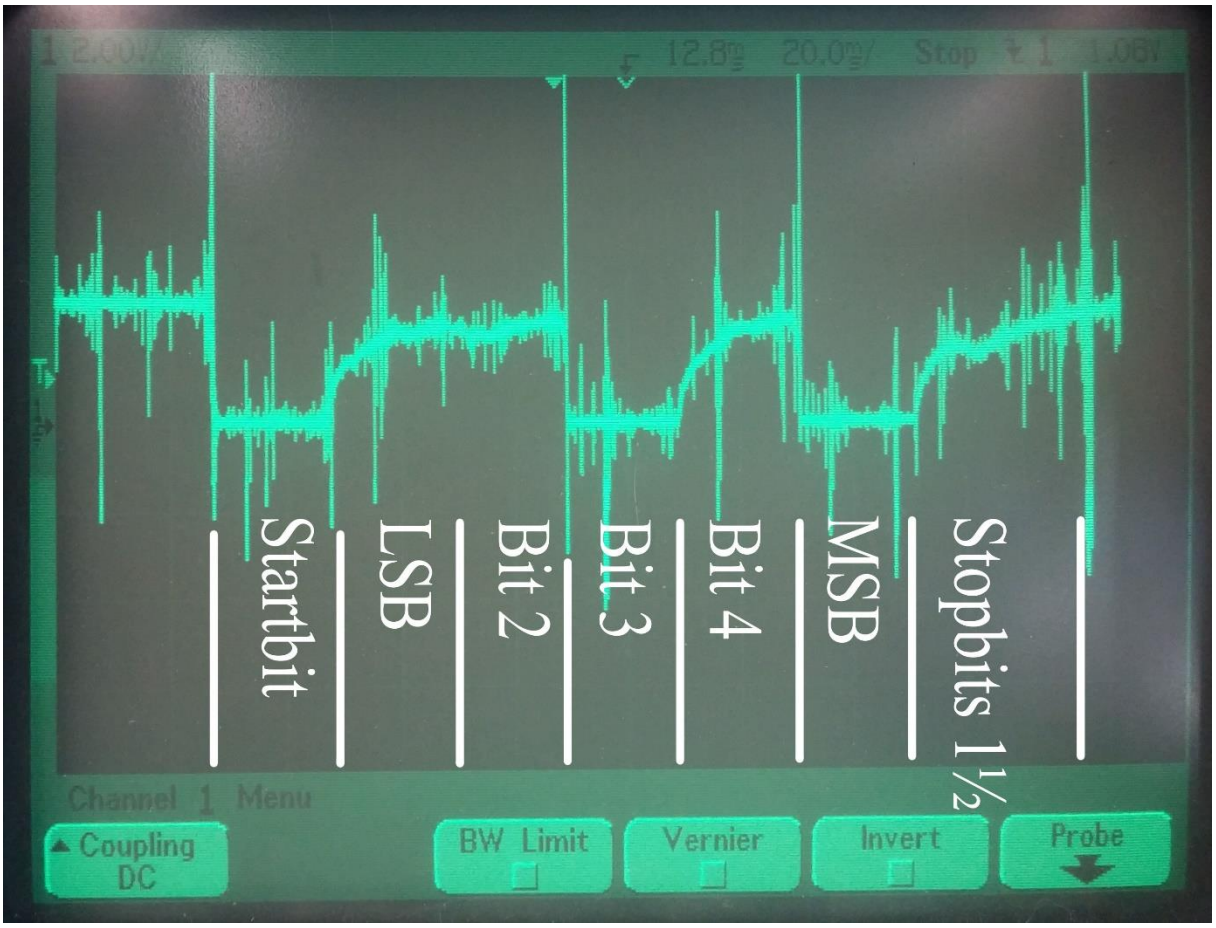

Abbildung 11 Signal nach der Reinigung, ein S wurde übertragen

Die Lötleiste, an der die Bananenbuchsen für den Signalpfad befestigt waren, wurde von uns durch eine gefräste Platine ersetzt, dabei wurde die Leitungsführung verschönert:

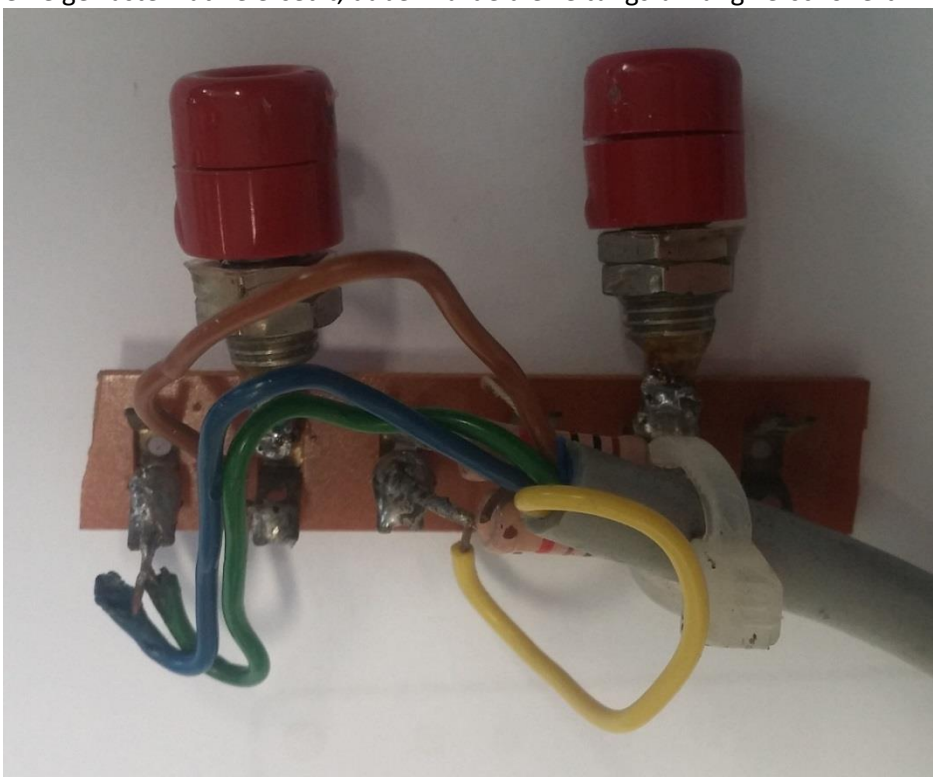

Abbildung 12 Lötleiste mit Bananenbuchsen

Abbildung 12 zeigt die vorherige Lötleiste, man erkennt hinter der Leitung die zwei bereits erwähnten 1 kΩ Widerstände und auf der linken Seite, dass die blauen und grünen Leiter verbunden sind.
Auf Abbildung 13 ist die neue Platine zu sehen, welche für diesen Zweck von uns hergestellt wurde. Auf ihr befinden sich lediglich vier parallele 2 kΩ Widerstände, vier Bohrungen mit Kontaktflächen, zwei für die Bananenbuchsen und zwei für die Leitungen, die in den Fernschreiber hineinführen. Außerdem sind noch 4 Bohrungen vorhanden, um Schrauben zu befestigen, die als Füße dienen. Die vorher vorhandene Brücke wurde ins Innere des Fernschreibers verlegt.

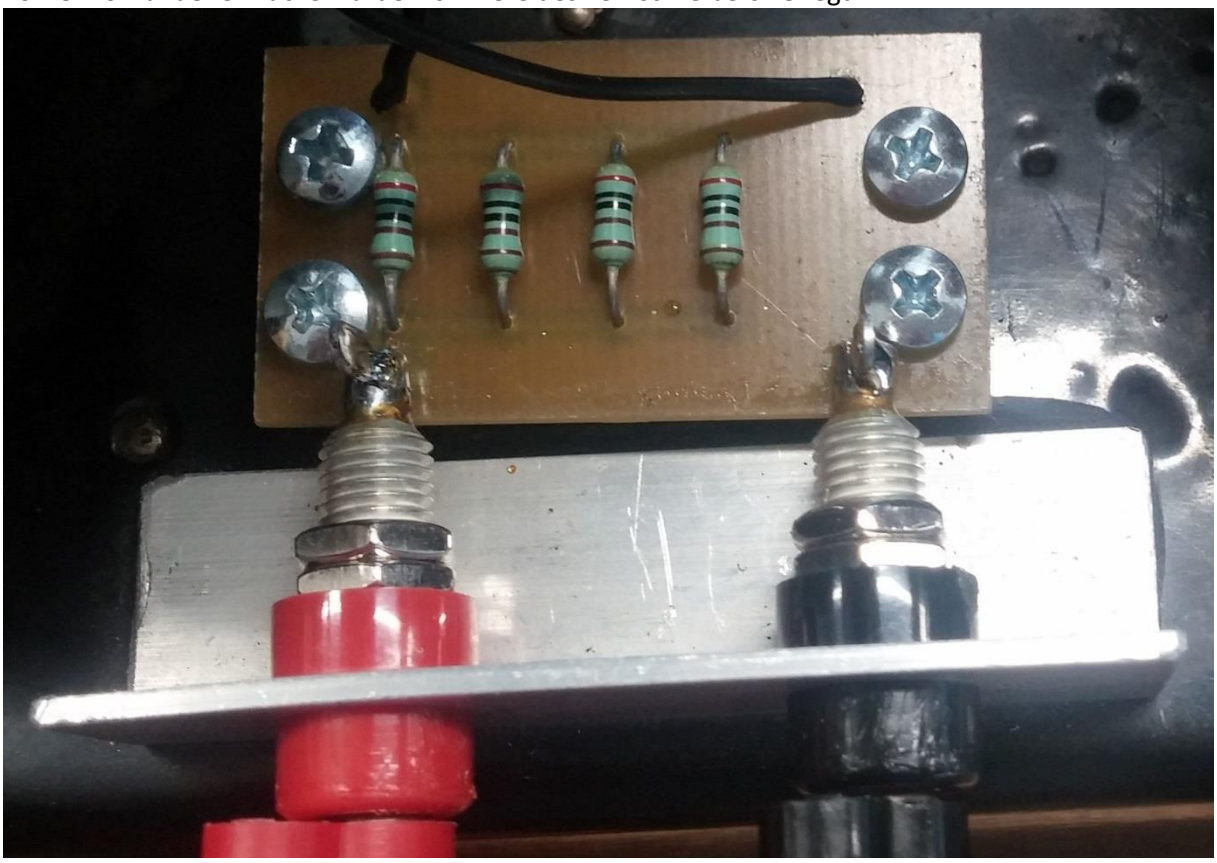

Abbildung 13 Neue Platine für die Anschlussbuchsen

Außerdem wurden einige Schrauben durch moderne ersetzt, da die alten Schrauben einen sehr schmalen Schlitz hatten.

5 Das Protokoll

Der Fernschreiber verwendet ein Protokoll sehr ähnlich zu V.24 bzw. RS-232 im Direktbetrieb, also zwischen zwei Datenendeinrichtungen, ohne Datenübertragungseinrichtungen dazwischen. Im Folgenden werden Gemeinsamkeiten und Unterschiede zwischen dem Protokoll des Fernschreibers und RS-232 dargelegt, für eine vollständige Beschreibung des RS-232 Protokolls wird Fachliteratur empfohlen[3], als Startpunkt für nähergehende Beschäftigung werden die Praktikumsunterlagen zum Praktikum Digitaltechnik von Herrn Rabe empfohlen[4].

5.1 Technische Daten

Die serielle Schnittstelle des Fernschreibers verwendet keinerlei Handshake-Leitungen und ermöglicht diese auch nicht, im Gegensatz zu RS-232. Ein weiterer großer Unterschied ist, dass der Fernschreiber eine Stromschleife verwendet, welche richtungsunabhängig einen Pegel von 0-40 mA hat. Das RS-232 Protokoll verwendet eine symmetrische Spannung von -15 V bis 15 V mit einer verbotenen Zone von -3 V bis 3 V.

Beide Protokolle arbeiten asynchron mit einer Wortbreite (Anzahl an Bits pro Übertragung) von 5 Bit (bei RS-232 variabel) und 1,5 Stoppbits (bei RS-232 1, 1,5 oder 2 möglich). Ebenfalls sind beide Protokolle low-aktiv (ein hoher Pegel entspricht einer logischen Null und umgekehrt).

Aufgrund der Tatsache, dass der Fernschreiber nur eine Stromschleife besitzt, ist kein Duplexbetrieb (gleichzeitiges Senden und Empfangen) möglich, RS-232 unterstützt auch den Vollduplexbetrieb. Das RS-232 Protokoll ermöglicht verschiedene Übertragungsraten, wobei Sender und Empfänger sich auf eine Übertragungsrate einigen müssen, damit die Kommunikation möglich ist. Bei RS-232 liegt die Übertragungsrate heutzutage etwa zwischen 2.400 Baud (Bit pro Sekunde) und 115.200 Baud, der Fernschreiber arbeitet mit 50 Baud.

5.2 Die Übertragung

Abbildung 14: Beispiel einer Übertragung an den Fernschreiber, schematisch dargestellt

In Abbildung 14 ist eine Übertragung dargestellt wie sie vom Einplatinencomputer aus vorgenommen wird. In Grün dargestellt ist der Ruhepegel. In Blau die Start- und Stoppbits und in Gelb das eigentliche Zeichen. Diese Darstellung findet sich im Betrieb auf dem Display wieder, näheres dazu im Abschnitt „Software“. Es werden zwei Stoppbits verwendet, da der UART des Einplatinencomputers nur ein oder zwei Stoppbits beherrscht. Das übertragene Zeichen ist ein H.

[3] Beispielsweise Joe Campbell: V *24 / RS-232 Kommunikation.*

[4] „Serielle Datenübertragung am Beispiel der V.24/RS-232-Schnittstelle“ von Prof. Dr.-Ing. Dirk Rabe, abrufbar unter http://www.technik-emden.de/~rabe/Downloads/digi/praktikum/PraktDigi_V1_V24.pdf

Um die Übertragung eines Zeichens einzuleiten wird ein Startbit gesendet, welches der Empfänger erkennt. Das codierte Zeichen wird mit dem LSB zuerst und low-aktiv übertragen. Die eigentliche Codierung für das H in high-aktiv und mit dem MSB zuerst ist also 01011. Am Ende der Übertragung werden Stoppbits übertragen, um das Ende eines Zeichens zu signalisieren und das synchronisieren zu vereinfachen. Die Möglichkeit 1,5 Stoppbits zu verwenden besteht im Protokoll aus historischen Gründen, da gerade bei Fernschreibern die Zeitgeber nicht sehr genau sind. Wenn man dann 1,5 Stoppbits verwendet, kann man trotzdem sicherstellen, dass sich die beiden beteiligten Geräte synchronisieren können, da eindeutig zu erkennen ist, dass die Zeit zu keinem Datenbit gehören kann.

5.3 Die Codierung

Übertragungscodierung Baudot, Engl. Version[5] [2]		Bits					Übertragungscodierung Fernschreiber	
Buchstaben	Zeichen	LSB	1	2	3	MSB	Buchstaben	Zeichen
A	1	1	0	0	0	0	A	:
B	8	0	0	1	1	0	B	?
C	9	1	0	1	1	0	C	(
D	0	1	1	1	1	0	D	,
E	2	0	1	0	0	0	E	3
/	[1]/	1	1	0	0	0		
F	[5]/	0	1	1	1	0	F	/
G	7	0	1	0	1	0	G	&
H	[1]	1	1	0	1	0	H	+
I	[3]/	0	1	1	0	0	I	8
J	6	1	0	0	1	0	J	Klingel
K	(	1	0	0	1	1	K	§
L	=	1	1	0	1	1	L	=
M	)	0	1	0	1	1	M	'
N	£	0	1	1	1	1	N	-
O	5	1	1	1	0	0	O	9
P	+	1	1	1	1	1	Löschen	Löschen
Q	/	1	0	1	1	1	Q	1
R	-	0	0	1	1	1	R	4
S	[7]/	0	0	1	0	1	S	Zeichen für Mark
T	[2]	1	0	1	0	1	T	5
U	4	1	0	1	0	0	U	7
V	'	1	1	1	0	1	V	)
W	?	0	1	1	0	1	W	2
X	[5]/	0	1	0	0	1	X	%
Y	3	0	0	1	0	0	Y	6
Z	:	1	1	0	0	1	Z	.
-	.	1	0	0	0	1		
Löschen	Löschen	0	0	0	1	1	P	0
Zeichen		0	0	0	1	0	Zeichen	
Buchstaben		0	0	0	0	1	Buchstaben	

Tabelle 1 Vergleich Baudot-Code und Code des vorliegenden Fernschreibers

Tabelle 1 lässt erkennen, dass nahezu alle Codepositionen doppelt belegt sind. Dies ist nötig, da nur mit 5 Bit übertragen wird. Damit lassen sich also 32 Zeichen codieren, wovon 31 Codepositionen auch genutzt werden (frei ist der Code „00000“, welcher nicht zuverlässig erkennbar wäre). Die 31 Positionen werden also doppelt belegt, dazu gibt es jeweils einen Code für die Umschaltung zum Buchstaben- oder Zahlenmodus. Welcher Modus gerade aktiv ist, wird entweder mechanisch oder

[5] Andere Darstellung auch unter http://www.galleyrack.com/images/artifice/telegraphy/tty/codes/baudot-alphabetic-full-code-2048x3440-w.jpg, bzw. http://www.circuitousroot.com/artifice/telegraphy/tty/codes/, abgerufen am 20.09.2016

digital gespeichert. Wird die Taste, welche in den gerade aktiven Modus wechseln soll betätigt, stellt sie ein Leerzeichen dar.
In Tabelle 1 ist außerdem zu erkennen, dass die meisten Buchstaben zwischen dem originalen Baudot-Code (auch CCITT-1 genannt) und der Codierung des Fernschreibers gleich sind, lediglich P und Löschen sind vertauscht. Bei den Zeichen weichen die beiden Codierungstabellen stärker voneinander ab, zum Beispiel kommen beim vorliegenden Fernschreiber die Zähler mit dem Bruchstrich nicht vor, dafür wurden andere Zeichen ermöglicht, z.B. ein Zeichen für die Klingel. Außerdem sind beim vorliegenden Fernschreiber 2 Codepositionen nicht belegt, die beim originalen Baudot-Code belegt sind.
Überprüft wurde die Codierung zunächst durch Messungen mit dem Oszilloskop über einen 100 Ω Widerstand und Tastaturanschlägen, bis wir feststellten, dass eine Mechanik an der Seite der Tastatur die Codierung ebenfalls darstellt, wie in Abbildung 15 gezeigt.

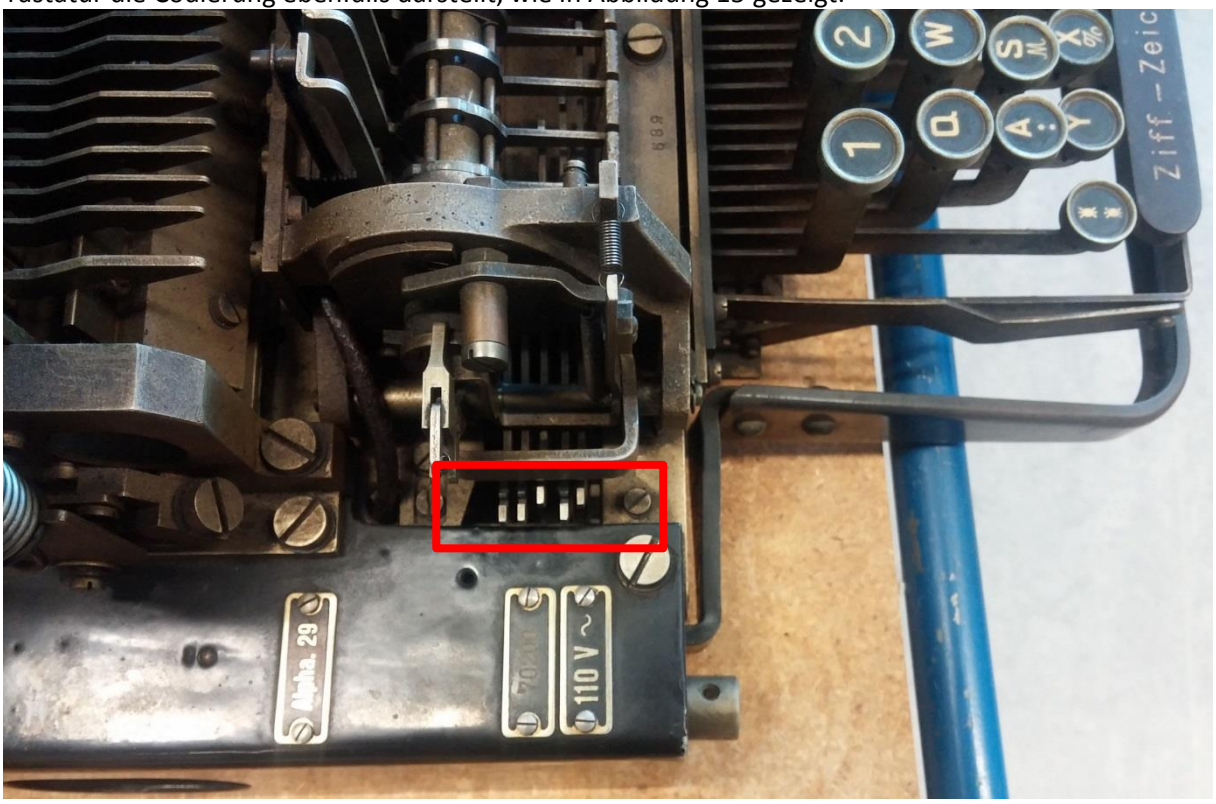

Abbildung 15 Fernschreiber von der Seite, Mechanik die die Codierung zeigt rot markiert

Dies wurde durch zusätzliche Messungen mit dem Oszilloskop überprüft und dann aufgrund geringerer Lärmbelastung (funktioniert auch wenn der Fernschreiber aus ist) als bevorzugte Methode gewählt.
Es ist wieder ein H dargestellt, aus dieser Perspektive ist also das LSB ganz links und weiter in Richtung Plaketten herausstehende Stifte stehen für eine eins.

6 Der Raspberry Pi

Der Raspberry Pi ist ein Einplatinencomputer, dessen besonderes Merkmal GPIO (General Purpose Input/Output)-Pins sind. Im Zuge dieser Projektarbeit wurde der Raspberry Pi 3 Model B, dargestellt in Abbildung 16 verwendet, welcher im Vergleich zu Vorgängern einen stärkeren Prozessor verwendet und Bluetooth sowie WLAN mit an Bord hat [3].

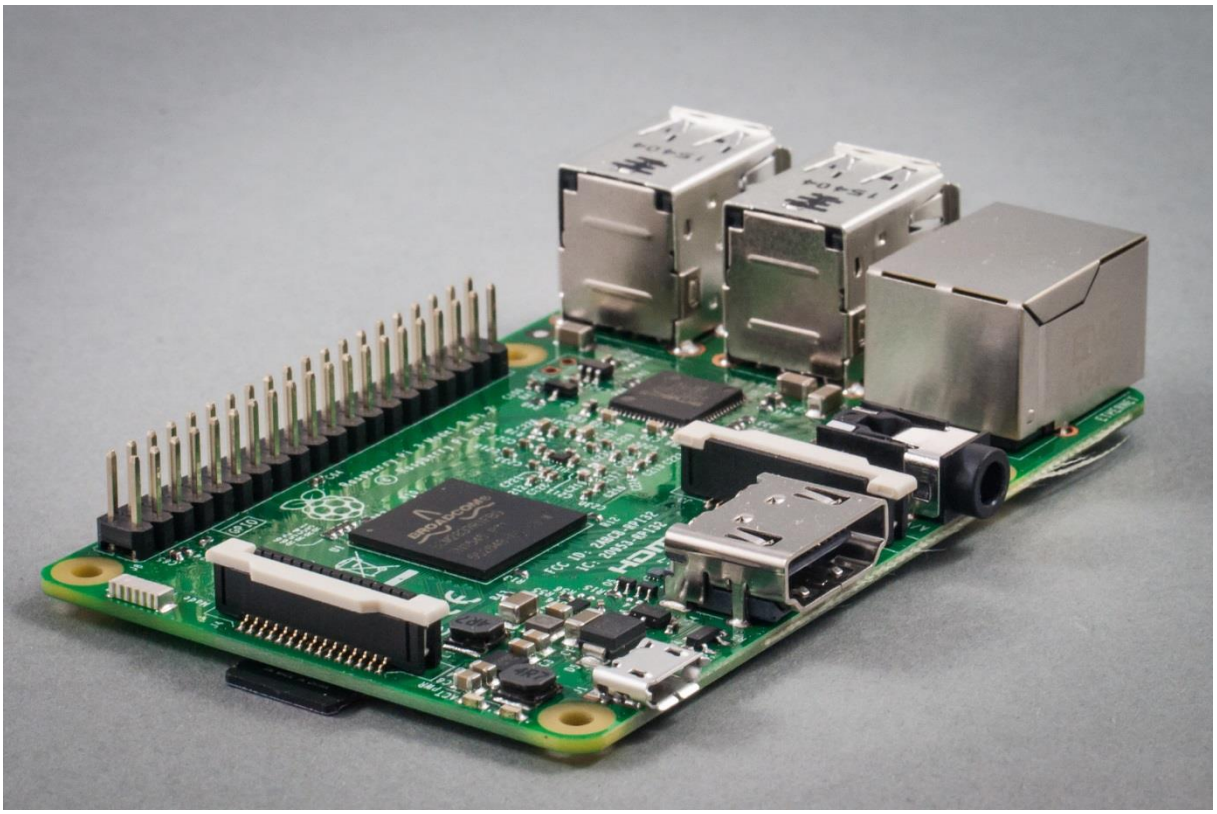

Abbildung 16 Der Raspberry Pi 3 Model B [4]

Die Maße des Raspberry Pis sind 93 mm mal 63,5 mm.
Versorgt wird der Raspberry Pi mit 5 V Betriebsspannung, die meist über ein Steckernetzteil der Mikro-USB Buchse zugeführt wird und dort mit einer 2,5 A PTC-Sicherung abgesichert ist [5][6].
An Anschlüssen besitzt der Raspberry Pi 3B 4x USB 2.0, 1x Fast Ethernet, wobei USB und Ethernet über einen gemeinsamen USB-Hub an den Prozessor angebunden sind [6], HDMI, einen kombinierten 3,5 mm Audioanschluss mit Composite Video Chinch Anschluss, ein Kamera-Interface (CSI), ein Display-Interface (DSI), einen Micro-SD-Karteneinschub, sowie 40 GPIO Pins [7].
Von besonderer Wichtigkeit bei der Projektarbeit war der UART (Universal Asynchronous Receiver/Transmitter), der in den Raspberry Pi integriert ist und über die GPIO-Pins zur Verfügung gestellt wird.

Ein UART ist eine elektronische Schaltung, die verwendet wird, um digitale serielle Schnittstellen zu realisieren. Viele Mikrocontroller und SoCs (System on a Chip) haben diese bereits eingebaut, so auch der SoC BCM2837, der im Raspberry Pi 3B (RPi 3B) verwendet wird. Die Schaltung übernimmt die Wandlung des seriellen Datenstroms in einen parallelen, hat meist Buffer verbaut und erkennt selbstständig die einzelnen Übertragungen. Dies erleichtert die Umsetzung der seriellen Schnittstelle ungemein, da man sich nicht mehr mit Problemen wie der Abtastung von Signalen und der Wandlung

[6] Das relevante Bauteil in der Quelle ist MF-MSMF250/X.

von seriellen in parallele Daten beschäftigen muss. Außerdem ist es dadurch möglich ein nicht-echtzeitfähiges Betriebssystem zu verwenden um eine serielle Verbindung aufzubauen.

Wie bereits erwähnt, ist eine der Besonderheiten des RPi 3B die integrierte Bluetooth-Schnittstelle. Das Problem, das sich uns dadurch bot, war, dass der Hardware-UART (UART0) für Bluetooth verwendet wurde und an den GPIO-Pins nur ein Mini-UART (UART1) bereitstand. Allerdings ist die Baudrate von UART1 von der verwendeten CPU-Frequenz, von der Belastung der CPU, Temperatur und weiterem abhängig und damit nicht verlässlich genug. Außerdem ist in UART1 nicht alles implementiert, was in UART0 zur Verfügung steht, zum Beispiel sind nicht alle Baudraten möglich und die Anzahl der Bits pro Wort sind nicht unterhalb von 6 wählbar. Als minimal einstellbare Baudrate haben wir 300 Baud festgestellt, somit war UART1 für uns nicht nutzbar.
Nach einiger Recherche haben wir Anleitungen gefunden[7], um die beiden UARTs zu tauschen und somit UART0 wieder an die GPIO-Pins zu binden. UART1 wird dann für Bluetooth verwendet. Somit war es auch möglich die vom Fernschreiber benötigten 50 Baud und 5 Bits pro Wort einzustellen. Da 1,5 Stoppbits nicht möglich sind, verwenden wir 2 Stoppbits, wodurch der Fernschreiber nur minimal in seiner Funktion beeinträchtigt wird. Das Empfangen von Daten wird um die Zeit für ein halbes Bit verlangsamt. Beim Senden von Daten darf nicht zu schnell hintereinandergeschrieben werden, da sonst das Stoppbit vom Raspberry Pi nicht richtig erkannt wird.

Auf den RPis wird Raspbian verwendet, welches auf Debian basiert. Die Auswahl an Betriebssystemen für die RPis ist groß[8], und im Grunde würde jedes Betriebssystem auf dem Raspberry Pi laufen, welches für die ARMv8-A-Architektur (beim RPi 3B) geeignet ist. Unsere Entscheidung für Raspbian wurde aufgrund einiger Vorteile getroffen:

1. Es ist das offiziell unterstützte Betriebssystem, somit ist die Verfügbarkeit und Aktualität von Treibern sichergestellt und es gibt offizielle Unterstützung
2. Außerdem ist es ein sehr verbreitetes Betriebssystem für den RPi, somit ist es unwahrscheinlich, dass wir die ersten wären, die ein bestimmtes Problem haben, was die Fehlersuchen im Verlauf des Projektes sehr erleichtert hat.
3. Zudem hat uns die grafische Oberfläche unterstützt, da wir dadurch die Raspberry Pis als Entwicklungsumgebung benutzen konnten und somit Übertragen von Dateien, sowie das Kompilieren für eine fremde Plattform entfallen ist.

Der Raspberry Pi 3 Model B wurde als Grundlage für unser Projekt gewählt, da er leicht zu programmieren ist, eine gute Schnittstelle zwischen Soft- und Hardware darstellt und gut verfügbar ist. Außerdem ist er weit verbreitet, was gute Unterstützung durch Anleitungen und bereits bestehende Problemlösungen, sowie gute Dokumentation bedeutet. Es wurde keine frühere Version der Raspberry-Familie verwendet, da der RPi 3B bereits WLAN an Bord hat, was Komplikationen mit Treibern und zusätzlichen Hardwareaufwand vermeidet und es uns ermöglicht hat, Funktionen umzusetzen, welche sonst nicht möglich gewesen wären.
Da der RPi 3B die aktuellste Version der Raspberry-Familie ist, ist auch die Softwareunterstützung mit aktueller Software für einige Zeit sichergestellt.

[7] z.B. http://www.fhemwiki.de/wiki/Raspberry_Pi_3:_GPIO-Port_Module_und_Bluetooth
[8] Hier die offiziell aufgeführten: https://www.raspberrypi.org/downloads/

7 Die Software

In diesem Bericht soll nicht im Fokus stehen, das gesamte von uns geschriebene Programm zu beschreiben, sondern beachtenswerte Abschnitte an Quelltext zu behandeln, von denen vielleicht manche etwas lernen können. Für alles Weitere sei auf die dem Bericht beiliegende CD/DVD verwiesen, auf der der gesamte, gut kommentierte Quelltext zu finden ist.

7.1 Erläuterung der verschiedenen Modi

In unserem Programm gibt es 3 verschiedene Modi, welche je nach Kontext angeboten werden. Um zu überprüfen welche Modi angeboten werden, wird in unserer Software überprüft, ob ein High-Pegel am RX-Pin anliegt, was der Fall ist, wenn der Stromkreis mit dem Fernschreiber geschlossen ist.
In Modus 1 und Modus 2 ist der RPi direkt mit dem Fernschreiber verbunden. Modus 1 stellt eine Kommunikation direkt zwischen dem Fernschreiber und dem RPi her, man kann also per angeschlossener Tastatur Text an den Fernschreiber übertragen und auch den vom Fernschreiber gesendeten Text direkt auf dem Display lesen.
Modus 2 stellt den Raspberry Pi als Webserver zur Verfügung. Wodurch man über ein Web-Interface oder einen weiteren Raspberry Pi, welcher dann in Modus 3 ist, mit dem Fernschreiber kommunizieren kann.
In Modus 3 erfolgt der Zugriff auf den direkt mit dem Fernschreiber verbundenen RPi in Modus 2, wodurch eine drahtlose Kommunikation mit dem Fernschreiber bereitgestellt wird.
In Modus 1 und Modus 2 kann die Kommunikation zusätzlich noch über einen angeschlossenen Thermodrucker ausgedruckt werden. Diese Möglichkeit wird ebenfalls auf dem Display angezeigt und eine Auswahl gefordert.

7.1.1 Modus 1 – Direkte Kommunikation

Modus 1 ist verfügbar, wenn der Fernschreiber angeschlossen ist. Dieser Modus dient dazu, direkt zwischen unserem Raspberry Pi und dem Fernschreiber zu kommunizieren, dazu werden die von uns gebaute Hardware und eine Tastatur benötigt.

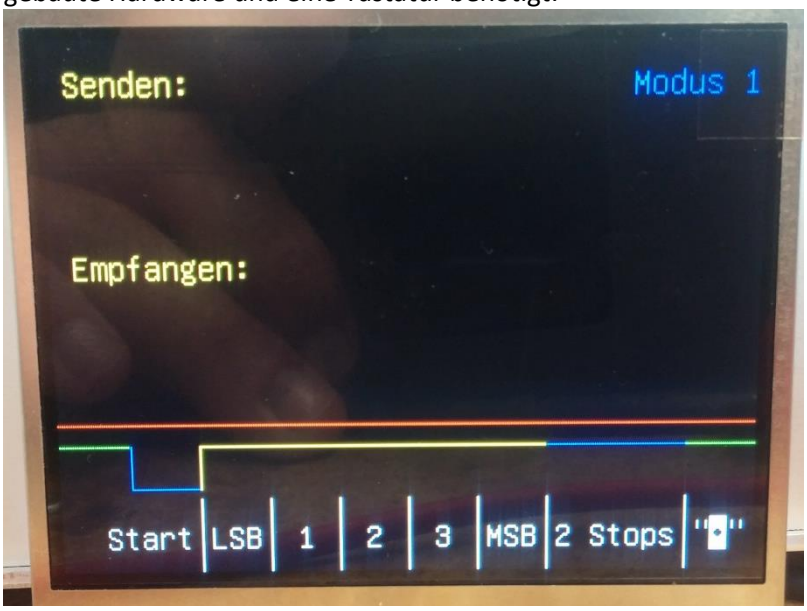

Abbildung 17 Displaydarstellung Modus 1

In Abbildung 17 sieht man die Darstellung auf dem Display für Modus 1. In der oberen rechten Ecke ist gekennzeichnet, in welchem Modus man sich befindet. Links davon ist das Feld für Senden, in dem bereits gesendetes, sowie der Text, der als nächstes gesendet werden soll, sichtbar ist. Dieses Feld hat insgesamt drei Zeilen, zwei davon für bereits gesendeten Text und eine Zeile für den Text, der als nächstes gesendet werden soll. Die Zeilenlänge ist auf 35 Zeichen begrenzt.

In der ersten Zeile steht der älteste Text, in der zweiten der Text, der zuvor gesendet wurde und in der Dritten der Text, der per Tastatur eingegeben wird um ihn zu senden. Wenn etwas gesendet wird, werden die drei Zeilen jeweils nach oben verschoben, womit der älteste Text verschwindet.
Darunter ist das Feld für das Empfangen, welches ebenfalls drei Zeilen enthält und auf dieselbe Weise aufgebaut ist. Allerdings hat der Fernschreiber keine Zeichen für Zeilenvorschub oder ähnliches, daher rückt die Zeile immer erst nach oben, wenn sie gefüllt wird.
Ganz unten, unterhalb des roten Striches ist schematisch die Übertragung des letzten Zeichens dargestellt, in Grün der Ruhepegel, in Blau Start- und Stoppbits und in Gelb die Datenbits. Unterhalb sind noch die einzelnen Bits gekennzeichnet und welches Zeichen dargestellt ist. Da „00000“ kein druckbares Zeichen ist, wird der schwarze Punkt auf weißem Hintergrund angezeigt.

Abbildung 18 Modus 1 in Gebrauch

Abbildung 18 zeigt Modus 1 während des Gebrauchs. Im Moment des Bildes wurde gerade etwas gesendet. Wenn man etwas zum Senden eingibt, erscheint der Text weiß. Wenn die Zeile komplett gefüllt ist, oder man die Eingabetaste drückt, werden die Zeichen nacheinander verschickt, wobei die bereits gesendeten Zeichen grün markiert werden. Die Zeilen mit dem bereits gesendeten Text sind ebenfalls grün. Da der Text noch nicht ganz verschickt wurde, ist das letzte Zeichen noch das letzte Empfangene, dies ändert sich, wenn die gesamte Zeile verschickt wurde.
Es können auch Befehlscodes eingegeben werden, die dann nicht an den Fernschreiber gesendet werden, dazu aber in Abschnitt 7.1.4-Befehlscodes mehr.

Im Empfangen-Feld sieht man bereits empfangenen Text. Dort sieht man auch, dass der Fernschreiber, wenn man zu schnell tippt, nicht immer die korrekten Zeichen sendet. Währenddessen schreibt der Fernschreiber alles auf Papierstreifen wie auf Abbildung 19 zu sehen.

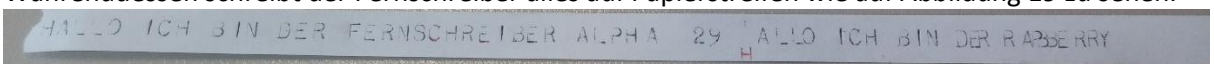

Abbildung 19 Vom Fernschreiber beschrifteter Papierstreifen

Für das Empfangen wurde ein Interrupt eingerichtet, der wartet bis ein GPIO Pin eine fallende Flanke signalisiert. Dazu kann nicht der RX-Pin selbst genommen werden, da dieser durch den UART benutzt wird. Daher wurde ein Pin direkt daneben genommen und die beiden Pins wurden parallelgeschaltet. Nachdem ein Interrupt erkannt wird, wird dieser Interrupt erst einmal deaktiviert, damit sich die Interrupts nicht gegenseitig blockieren. Dann wird der gesamte Buffer vom UART ausgelesen und es wird überprüft, ob der Inhalt noch in die jetzige Zeile passt. Wenn ja, wird der Inhalt des Buffers in die Zeile geschrieben, wenn nicht, werden die Zeilen im Empfangen Feld nach oben geschoben und eine neue Zeile begonnen.

Das Display wird nun aktualisiert, der Text wird gedruckt, wenn man dies am Anfang ausgewählt hat, und die Zeilen werden in Dateien geschrieben, was in diesem Modus allerdings nicht relevant ist. Als letzter Schritt wird der Interrupt wieder aktiviert.
Im Hintergrund wird auf Zeichen gewartet, wenn gerade keine Zeichen empfangen werden. Wenn ein Zeichen eingegeben wird, wird das Display aktualisiert, um dieses Zeichen darzustellen und es wird in einem Array gespeichert. Wenn das Zeichen ein Backspace ist, wird das letzte eingegebene Zeichen gelöscht, bei Enter wird die Zeile losgeschickt, sofern die Zeile keinem Befehl entspricht. Bei Erreichen der maximalen Zeilenlänge wird die Zeile ebenfalls losgeschickt.
Zum Senden wird jedes einzelne Zeichen an den UART übergeben, dann wird gewartet, bis das Zeichen abgeschickt wurde, dieses wird dann auf dem Display grün gefärbt und das nächste Zeichen wird abgeschickt, bis keine mehr vorhanden sind. Wenn man es so gewählt hat, wird die Zeile gedruckt und zum Schluss werden alle Zeilen im Senden-Feld nach oben geschoben.
Um nicht das zu empfangen, was man gesendet hat, wird während des Sendens der Interrupt deaktiviert und am Ende wird der Inhalt des Eingangsbuffers vom UART verworfen und der Interrupt wieder aktiviert.

7.1.2 Modus 2 – Betrieb als Webserver

Modus 2 ist ebenfalls nur verfügbar, wenn der Fernschreiber angeschlossen ist. Dieser Modus dient dazu, den am Fernschreiber angeschlossenen Raspberry als Server zu benutzen. Um den Raspberry als Server bereit zu stellen, erzeugt dieser auch sein eigenes WLAN. Die Kommunikation mit dem Fernschreiber läuft dann über die vom Server bereit gestellte Webseite oder einen zweiten Raspberry, der in Modus 3 läuft.

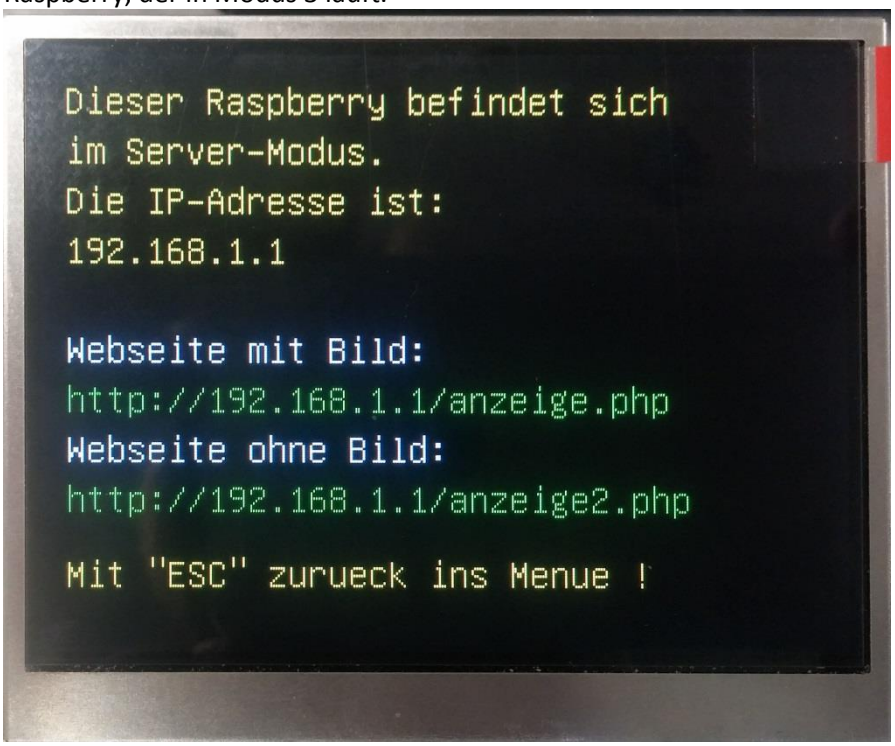

Abbildung 20 Displaydarstellung Modus 2

Über die Webseite oder den anderen Raspberry wird eine Datei auf dem Server geändert, die von dem Server immer wieder ausgelesen wird und bei neuem Inhalt dann versendet wird. Der Server schreibt im Gegenzug Text in drei verschiedene Dateien: eine für den bereits gesendeten Text, eine für den empfangenen Text und eine Datei für das letzte übertragene Zeichen. Die drei Dateien können dann von der Webseite bzw. dem anderen RPi ausgegeben werden.
Auf dem Bildschirm sichtbar sind in diesem Modus nur die IP-Adresse des Servers und die Website, die benutzt werden soll, um mit dem Fernschreiber zu kommunizieren (siehe Abbildung 20).
Um diesen Modus zu verlassen, benutzt man einen der Befehlscodes bei einem der Clienten oder drückt bei der Tastatur, die am Server angeschlossen ist, ESC.

Wenn man mit einem Smartphone, Tablet oder Laptop diesen Modus verwenden möchte, muss man im selben Netzwerk wie der Server sein, dazu stellt der Raspberry Pi ein WLAN-Netzwerk zur Verfügung, in welches man sich einloggen muss. Die SSID dieses Netzwerks ist „Fernschreiber“, das Passwort ist ebenfalls „Fernschreiber“.

7.1.3 Modus 3 – Client für den Webserver

Der Modus 3 wird angeboten, wenn kein Fernschreiber angeschlossen ist. Modus 3 ist der Client-Modus, wenn man einen Raspberry als Clienten für einen Raspberry in Modus 2 verwenden möchte. Um den Modus zu verwenden, muss man die IP-Adresse des Server-RPi angeben. Per ftp wird dann der zu sendende Text an den Server übertragen und die anderen, für die Anzeige relevanten Daten werden heruntergeladen.
In Abbildung 21 lässt sich erkennen, dass diese Darstellung nahezu identisch zu der Darstellung von Modus 1 ist. Dies ist zweckmäßig, da der Modus auch identisch verwendet wird. Der einzige Unterschied ist, dass der Raspberry, den man bedient, nun nicht mehr direkt mit dem Fernschreiber verbunden ist.

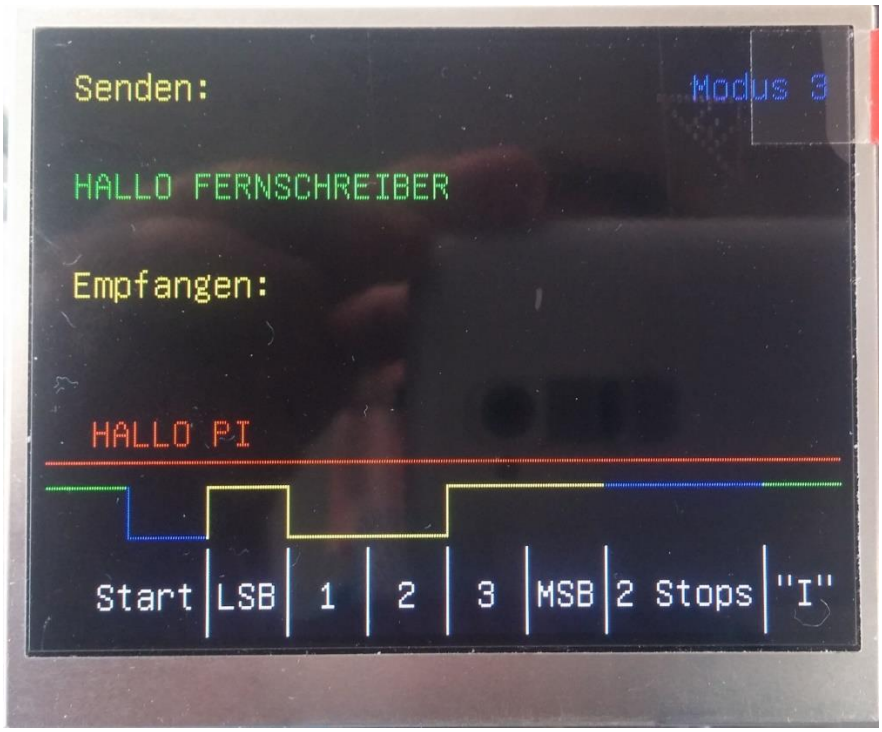

Abbildung 21 Modus 3 in Gebrauch

7.1.4 Befehlscodes

Befehl	Funktion	Modus		
		1	2	3
EXIT	Programm wird beendet (Nur zu Programmieranwendungen!)	x		
SHUTDOWN	Eigener Raspberry wird heruntergefahren	x		x
CHANGE	Moduswechsel	x		x
CLEAR	Display leeren (Senden/Empfangen)	x		x
CONNECT	Wenn Stromschleife angeschlossen Relais AN	x	x	x
DISCONNECT	Wenn Stromschleife angeschlossen Relais AUS	x	x	x
CHANGESERVER	Moduswechsel beim Server (und beim eigenen Raspberry)		x	x
SHUTDOWNSERVER	Server herunterfahren		x	x
"	Zeichen für den Buchstabenmodus übertragen	x	x	x
$	Zeichen für den Zahlenmodus übertragen	x	x	x
#	Zeichen für die Klingel übertragen	x	x	x

Tabelle 2 Übersicht über die Befehlscodes, die in den verschiedenen Modi verfügbar sind

In Tabelle 2 sind die Befehlscodes aufgeführt. Von links nach rechts ist in einer Zeile immer der Befehlscode, die Funktion die dieser auslöst und in welchem Modus er verfügbar ist aufgeführt. Wenn ein Befehl bei Modus 2 eine Markierung hat, bedeutet dies, dass man ihn auf der Webseite eingeben kann. Die Befehlscodes können nicht vom Fernschreiber aus verwendet werden.
Bis auf die letzten drei Befehle werden keine an den Fernschreiber übertragen. Für alle Befehlscodes außer die letzten drei gilt auch, dass nur die Befehle eingegeben werden dürfen, sonst wird der Befehl als normaler Text interpretiert und übertragen. Beispiel: „CLEAR“ geht, „CLEARabc“ geht nicht durch die zusätzlichen Buchstaben, selbst „ CLEAR“ wird durch das Leerzeichen vor dem Befehl als Text übertragen.
Wenn das Zeichen für die Klingel übertragen oder empfangen wird, klingelt auch der Raspberry Pi, welcher direkt mit dem Fernschreiber verbunden ist. Wenn dieser aber im Servermodus ist und ein weiterer Raspberry Pi im Modus 3 verbunden ist, zeigt der RPi im Modus 3 das Klingeln auf dem Display an. Auf der Weboberfläche ist das Klingeln nur als Zeichen zu sehen.

7.2 Ausgewählte verwendete Bibliotheken

Eine wichtige Bibliothek im Verlauf dieser Projektarbeit war WiringPi von Gordon Henderson[9]. Sie dient dazu die GPIO-Pins des Raspberry Pi anzusteuern und stellt weitere Funktionalitäten bereit, unter anderem eine Art Interrupt. Da wir auf dem Raspberry Pi Raspbian installiert haben, welches kein Echtzeitbetriebssystem ist und auf Linux basiert, gibt es keine Interrupts, welche man verwenden könnte, da der Kernel die Hardware komplett alleine verwaltet. Dadurch müsste man im Kernel etwas registrieren, der Kernel betreibt dann ein energiesparendes Polling und weckt das Programm bei einer registrierten Änderung auf. Dieser Prozess ist durch WiringPi wesentlich einfacher, man ruft im Grunde nur eine Funktion auf. Auch für Zeitverzögerungen und vieles weitere sind Funktionen vorhanden.

Die BCM2835-Bibliothek von Mike McCauley[10] erfüllt denselben Zweck wie die WiringPi-Bibliothek und musste von uns eingebunden werden, da die vom Hersteller des Displays bereit gestellten Bibliotheken die BCM2835-Bibliothek verwenden. In unserem eigenen Code wurde allerdings durchgängig WiringPi benutzt, da dies einfacher anzuwenden ist und Interrupts möglich macht.

Für das verbaute Display wurde vom Hersteller eine Bibliothek mit dem Namen RAIO8870 mitgeliefert[11]. Sie übernimmt die Initialisierung und Ansteuerung des für das Display verwendeten Display-Controllers RAiO RA8870 von RAiO TECHNOLOGY INC. und stellt Funktionen bereit, wie das Zeichnen von Text und verschiedenen geometrischen Formen.

[9] Verfügbar unter http://wiringpi.com/, abgerufen am 22.09.2016
[10] Verfügbar unter http://airspayce.com/mikem/bcm2835/index.html, abgerufen am 22.09.2016
[11] Verfügbar unter http://admatec.de/sites/default/files/downloads/C-Berry.tar.gz, abgerufen am 22.09.2016

7.3 Einrichtung der seriellen Schnittstelle auf dem Raspberry Pi

Um in unserem Programm die serielle Schnittstelle verwenden zu können, musste, wie bereits erwähnt, der UART mit dem Mini-UART getauscht werden. Nachdem dies geschehen ist, kann man im Programmcode die Parameter wie gewünscht setzen, mit dem Mini-UART waren nicht alle benötigten Parameter möglich und das Timing ist nicht konstant (siehe Abschnitt 6).

```
1.  int serialinit(void)
2.  {
3.        int uart0_filestream = -1;
4.
5.        uart0_filestream = open("/dev/serial0", O_RDWR | O_NOCTTY | O_NDELAY);
6.        if (uart0_filestream == -1)
7.        {
8.              printf("[ERROR] UART open()\n");
9.        }
10.
11.       struct termios options;
12.             tcgetattr(uart0_filestream, &options);
13.                   options.c_cflag = B50 | CS5 | CLOCAL | CREAD | CSTOPB;
14.                   options.c_iflag = IGNPAR;
15.                   options.c_oflag = 0;
16.                   options.c_lflag = 0;
17.             tcflush(uart0_filestream, TCIFLUSH);
18.             tcsetattr(uart0_filestream, TCSANOW, &options);
19.       return uart0_filestream;
20. }
```

Codeabschnitt 1 Initialisierung der seriellen Schnittstelle

In Codeabschnitt 1 ist dargestellt, wie die serielle Schnittstelle initialisiert wurde. Zunächst öffnet man die serielle Schnittstelle mit open() mit den Optionen O_RDWR (Schreib- und Lesezugriff), O_NOCTTY und O_NDELAY. O_NOCTTY sorgt dafür, dass die geöffnete Schnittstelle keine Kontrollsignale an den öffnenden Prozess senden kann und O_NDELAY ist das Flag, um etwas im nicht-blockierenden Modus zu öffnen, dadurch wartet der serielle Port nicht, wenn gelesen wird und nichts im Eingangsbuffer ist.
Wenn open() funktioniert, wird eine Nummer zurück geliefert, unter der der serielle Port angesprochen werden kann, bei einem Fehler wird -1 zurückgegeben und ein Fehler wird ausgegeben.
Um die Einstellungen der Schnittstelle zu ändern, wird ein struct benötigt, dessen Struktur in termios.h beschrieben ist. Um nicht alle Optionen selbst setzen zu müssen, werden zunächst mit tcgetattr() die bisherigen Einstellungen übernommen. Danach werden die gewünschten Einstellungen in die Flags geschrieben.
c_cflag ist zuständig für generelle Einstellungen der Kommunikation, hier wurde mit B50 die Baudrate auf 50 Baud gestellt, CS5 bedeutet, dass 5 Bits pro Zeichen verwendet werden, CLOCAL sorgt dafür, dass jegliche Kontrollleitungen (alles außer RX und TX), wenn sie vorhanden wären, ignoriert werden. CREAD ermöglicht das Lesen zusätzlich zum Schreiben und CSTOPB setzt die Anzahl der Stoppbits auf zwei.
c_iflag beinhaltet die Flags für das Lesen. Da der Fernschreiber kein Paritätsbit verwendet, wird der seriellen Schnittstelle mit IGNPAR gesagt, dass es keine Parität gibt.
Mit c_oflag kann man Dinge einstellen, die das Senden betreffen. Da wir davon nichts benötigen, werden alle Flags auf 0 gesetzt.
Wir verfahren mit c_lflag auf dieselbe Weise, da dies auch nur Dinge enthält, die wir nicht benötigen.
Danach wird mit tcflush() der Inhalt des Input-Buffers gelöscht und mit tcsetattr() werden die Optionen gesetzt.

7.4 Umwandlung von ASCII in Baudot und zurück

```
1  char tauschtx(unsigned char outchar)
2  {
3        unsigned char code[256];
4        int i;
5
6        if(outchar==' ' && zb==MODUS_BUCHSTABE)
7              return 0xEF;
8        else if(outchar==' ' && zb==MODUS_ZAHL)
9              return 0xF7;
10
11       for(i=0;i<7;i++)
12       {
13             code [i]= 0x11;
14       }
15
16       code [7]= 0xF6; //Klingel
17       code [8]= 0xFF; //Backspace
18       for(i=9;i<37;i++)
19       {
20             code [i]= 0x11;
21       }
22       code ['%']= 0xED;
23       code ['&']= 0xF5;
24       code ['\'']= 0xE5;
25       code ['(']= 0xF2;
26       code [')']= 0xE8;
27       code [42]= 0x11;
28       code ['+']= 0xF4;
29       code [',']= 0xF0;
         […]
30       code ['w']= 0xE9;
31       code ['x']= 0xED;
32       code ['y']= 0xFB;
33       code ['z']= 0xEC;    //122
34       for(i=123;i<245;i++)
35       {
36             code [i]= 0x11;
37       }
38       code ['§']= 0xE6;    //245
39       for(i=246;i<256;i++)
40       {
41             code [i]= 0x11;
42       }
43       code['\"']=0xEF;           //Buchstabenmodus schreiben
44       code['$']=0xF7;            //Zahlenmodus schreiben
45       code['#']=0xF6;            //BING!
46
47       return(code[(int)outchar]);
48 }
```

Codeabschnitt 2 Übersetzung von ASCII-Zeichen in Baudot-Code (Auszug)

In Codeabschnitt 2 ist auszugsweise die Umwandlung von ASCII-Zeichen in den Baudot-Code gezeigt. In zb wird global der derzeit aktive Modus gespeichert, da der Fernschreiber ebenfalls 2 verschiedene

Modi hat. In den Zeilen sechs bis neun ist zu sehen, wozu dies in dieser Funktion benötigt wird: Für die verschiedenen Modi gibt es verschiedene Leerzeichen. Kommt das andere Leerzeichen beim Fernschreiber an, wird es als Moduswechsel interpretiert. Sollte dies also aus Versehen passieren, ist der Text auf der anderen Seite nicht mehr lesbar. Wird beispielsweise die Taste „Buchstaben“ auf der Tastatur des Fernschreibers betätigt und der Fernschreiber befindet sich bereits im Buchstabenmodus, so stellt dieser Tastenanschlag ein Leerzeichen dar, ansonsten wird auf Buchstabenmodus umgeschaltet (siehe auch Abschnitt 5.3)
Die Umwandlung geschieht mithilfe eines Arrays, beim Fernschreiber nicht vorhandene Zeichen werden mit 0x11 belegt und später abgefangen, so dass sie gar nicht erst übertragen werden. Zusätzlich wurden einige vom Fernschreiber nicht benutzbare Zeichen als Steuerzeichen umfunktioniert. Ein Anführungszeichen (") überträgt das Zeichen für den Buchstabenmodus, wenn dieser schon aktiv ist, ist das für den Fernschreiber nur ein Leerzeichen. Bei dem Dollarzeichen wird dasselbe für den Zahlenmodus getan. Da es auf einer heutigen QWERTZ-Tastatur kein Zeichen für eine Klingel gibt, wurde dafür das Raute Zeichen (#) verwendet, da dieses nicht auf der Fernschreiber-Tastatur vorhanden ist.
Der Inhalt des Arrays ergibt sich aus der in Abschnitt 5.3 aufgeführten Tabelle. Dazu mussten die in der Tabelle angegebenen Werte invertiert werden, da der UART nicht low-aktiv überträgt. Da der UART allerdings das LSB automatisch von hinten nach vorne stellt, musste die Reihenfolge gegenüber der Tabelle geändert werden. Z.B. ist Z mit „11001“ in der Tabelle angegeben, wobei das LSB vorne steht. Mit dem MSB vorne wäre das „10011“ und invertiert „01100“. Da es ein char-Array ist, brauchen wir aber acht Bits für jeden Wert. Dazu wurde vor dem MSB mit drei Einsen aufgefüllt, weil der UART von hinten nach vorne überträgt, werden diese zusätzlichen Einsen vom UART dann ignoriert. Dadurch wird unsere Zahl dann „1110 1100“. Da man keine binären Zahlen in C-Quellcode angeben kann, ist eine Umwandlung in hexadezimale Zahlen zweckmäßig. Jedes Nibble (Halbbyte) entspricht einem Zeichen in Hexadezimal: „1110“ ist ein E und „1100“ ein C, zusammen macht das „0xEC“.
Da der Fernschreiber ausschließlich Großbuchstaben beherrscht, bekommen jeweils der Kleinbuchstabe und der Großbuchstabe dieselbe Zahl zugewiesen.
Insgesamt wird der Funktion ein ASCII-Zeichen als char übergeben und zurück geliefert wird eine char-Variable mit einer Zahl, welche der UART übertragen kann.

Die andere Richtung der Übersetzung wird von der Funktion tauschrx() übernommen:

```
1   char tauschrx(unsigned char inchar)
2   {
3     inchar= inchar | 0xE0;
4     unsigned char code[256];
5     int i;
6     for(i=0;i<224;i++)  //Diese koennen bei der Rueckumwandlung nicht vorkommen
7     {
8           code[i]=' ';
9     }
10    if (zb==MODUS_FEHLER && inchar == 0xF7)  //Wenn modus nicht definiert ist
11    {                                                          //-> Modus
                                                                 richtig setzen
12          zb=MODUS_ZAHL;                                       //& Funktion beenden
13          return 0x00;
14    }
15    else if(zb==MODUS_FEHLER && inchar==0xEF)
16    {
17          zb=MODUS_BUCHSTABE;
18          return 0x00;
19    }
20    else if (zb==MODUS_BUCHSTABE) //Buchstaben
21    {
22          code[0xE0]=' ';     //Frei
23          code[0xE1]='N';
            […]
24          code[0xF7]=0x00;    // auf Zahl umstellen
25          code[0xF8]='O';
26          code[0xF9]='I';
27          code[0xFA]='U';
28          code[0xFB]='Y';
29          code[0xFC]=' ';     //Frei
30          code[0xFD]='E';
31          code[0xFE]='A';
32          code[0xFF]=0x08;    //Backspace
33    }
34    else if (zb==MODUS_ZAHL) //Zahl
35    {
36          code[0xE0]=' ';     // Frei
37          code[0xE1]='-';
            […]
38          code[0xFE]=':';
39          code[0xFF]=0x08;
40    }
41    if (code[inchar]==0x00)
42    {
43          if (zb==MODUS_BUCHSTABE)          // Buchstabe
44          {
45                zb=MODUS_ZAHL;              // umstellen auf Zahl
46          }
47          else
48          {
49                zb=MODUS_BUCHSTABE;
50          }
51    }
52    return(code[inchar]);
53  }
```

Codeabschnitt 3 Übersetzung von Baudot-Code in ASCII-Zeichen (Auszug)

Codeabschnitt 3 zeigt, wie die vom Fernschreiber empfangenen Zeichen wieder in ASCII-Zeichen umgewandelt werden. Zunächst werden in Zeile 3 die ersten drei Bits des eingehenden Codes auf 1

gesetzt. In Zeile 6 bis 9 werden die ersten 223 Felder des Arrays code auf das Leerzeichen gesetzt, da diese nicht im weiteren Verlauf vorkommen können („11100000“ bzw. „0xE0“ ist 224, dies ist das erste Zeichen welches auftreten kann). In den Zeilen 10 bis 19 wird, sollte der Fernschreiber zuerst senden, der richtige Modus eingestellt und die Funktion beendet. Für die verschiedenen Modi gibt es verschiedene Tabellen, da die Zeichen auf der Fernschreiber-Tastatur größtenteils doppelt belegt sind.
Wenn ein Zeichen für einen Modus kommt, wird dies als Leerzeichen interpretiert, wenn der Modus gleich ist. Ansonsten wird dieses Feld auf 0x00 gesetzt, was dann im Folgenden in den Zeilen 41-51 für einen Moduswechsel sorgt. Am Ende (Zeile 52) wird dann das korrekte Zeichen an die aufrufende Funktion zurückgegeben.
Bitfolgen, welche auf der Fernschreiber-Tastatur nicht vorhanden sind, werden als Leerzeichen definiert, damit selbst bei Fehlern keine nicht-druckbaren Zeichen auf dem Display auftauchen.
Bevor wir die Hardware hergestellt haben, wurde die serielle Kommunikation im Baudot-Code zwischen zwei Raspberry Pis ausgiebig getestet.

7.5 Webinterface und Hotspot

Da man die Zugangsdaten für WLAN-Netzwerke in den Konfigurationsdateien des Raspberry Pi in Klartext finden kann, wurde eine Möglichkeit gesucht, bei Abschluss des Projekts nicht mehr mit dem eduroam-Netzwerk der Hochschule verbunden zu sein.
Nach ein wenig Recherche fanden wir die Möglichkeit, dass der Raspberry Pi selbst ein WLAN-Netzwerk bereitstellen könnte. Da der Raspberry Pi 3 Model B bereits eine WLAN-Antenne verbaut hat, war dies nur noch eine Sache der Konfiguration, welche dank Linux-Betriebssystem und den Hilfsprogrammen hostapd und dnsmasq kein Problem darstellt. hostapd dient dazu, einen Hotspot bereit zu stellen und die Authentifizierung unter anderem per WEP oder WPA vorzunehmen. dnsmasq dient hierbei als DNS-Server und weist IP-Adressen zu. Es wäre möglich, mit dem Raspberry Pi einen Hotspot für ein bestehendes Netzwerk aufzubauen, indem man ein Ethernet-Kabel anschließt und eine Netzwerkbrücke aufbaut, aber das war nicht nötig.

Die Webseite, die der Raspberry Pi zur Verfügung stellt, gibt es in 2 verschiedenen Versionen. Die erste (siehe Abbildung 22) enthält ein Textfeld, in das der zu sendende Text eingetragen werden kann. Klickt man auf den Absenden-Button, wird das zu Sendende per AJAX[12] an ein PHP-Skript übertragen, welches den Text in eine Textdatei schreibt.
Der gesendete und empfangene Text wird per AJAX alle 100 ms aus einer Textdatei ausgelesen, welche vom Serverprogramm geschrieben wird. Das Bild wird durch ein weiteres PHP-Skript erzeugt, welches alle 500 ms per AJAX aufgerufen wird.
Durch die Verwendung von AJAX funktioniert das gesamte Webinterface, ohne dass die Webseite komplett aktualisiert werden muss. Die Verwendung von AJAX basiert dabei hauptsächlich auf der JQuery-Bibliothek[13].

[12] Asynchronous JavaScript and XML, Möglichkeit der Asynchronen Datenübertragung für Webseiten
[13] https://jquery.com/

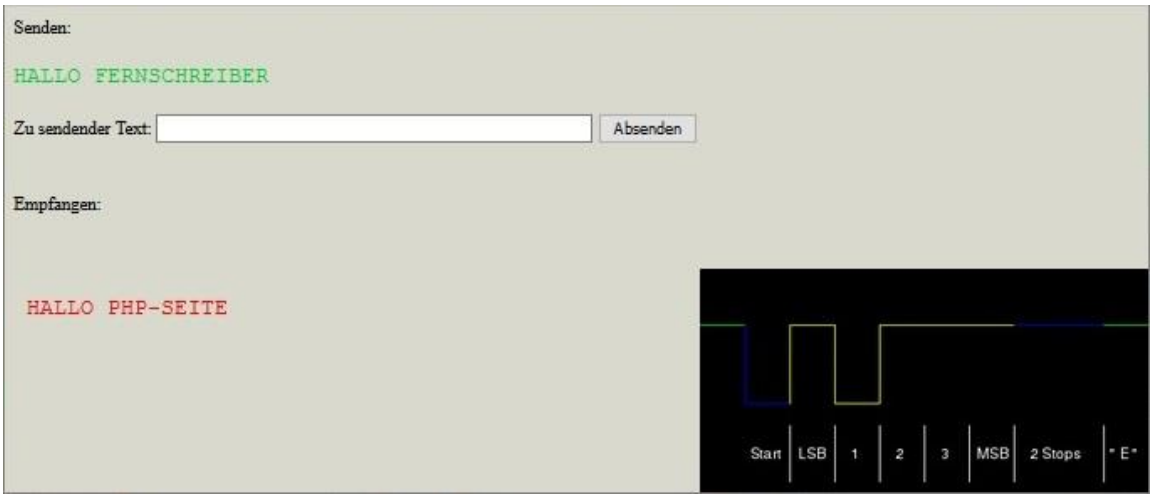

Abbildung 22 Inhalt des Webinterfaces mit Bild

Abbildung 23 Inhalt des Webinterfaces ohne Bild

Abbildung 23 stellt das Webinterface ohne Bild dar. Dies ist vornehmlich erstellt worden, um die Verwendung mit Smartphones übersichtlicher zu gestalten, da auf deren Bildschirmen nicht so viel Platz ist. Funktionsmäßig besteht kein Unterschied, außer, dass das Bild nicht aktualisiert wird und es unten noch einen Beenden-Button gibt, welcher den Tab/das Fenster schließt.
Die Webseite mit Bild ist unter [IP-Adresse des Servers]/anzeige.php zu erreichen, die ohne Bild unter [IP-Adresse des Servers]/anzeige2.php.

7.6 Display

Das verwendete Display ist ein C-Berry TFT-Display mit 3,5“ Größe von Admatec. Die Auflösung beträgt 320x240 Pixel und die Display-Hintergrundbeleuchtung basiert auf LEDs. Das Display wird per SPI angesprochen und beinhaltet einen RA8870 Grafikcontroller. Zur Verbindung mit dem Raspberry Pi dient eine Adapterplatine, die in Abbildung 24 dargestellt ist. Mit der Adapterplatine wird das Display per Flachbandkabel verbunden [8]. In Abbildung 25 ist das Display inkl. Flachbandkabel zu sehen.
Um die Programmierung des Displays zu vereinfachen, stellt der Hersteller des Displays ein Paket mit Beispielprogrammen, Bibliotheken und Dokumentation zur Verfügung. Mithilfe der Beispielprogramme konnten wir uns einarbeiten und dieses Wissen dann im weiteren Verlauf zur Programmierung nutzen.

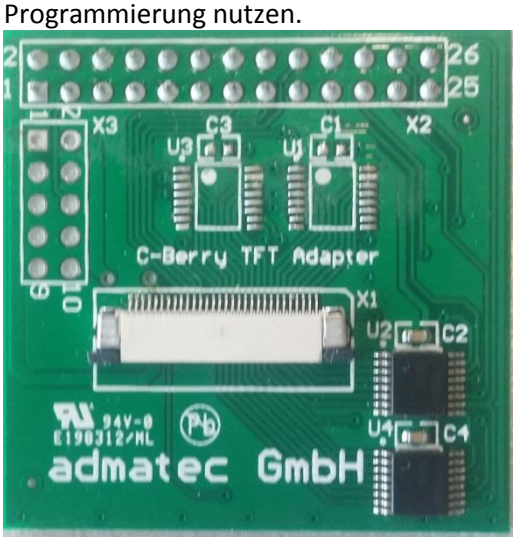

Abbildung 24 Adapterplatine für das Display

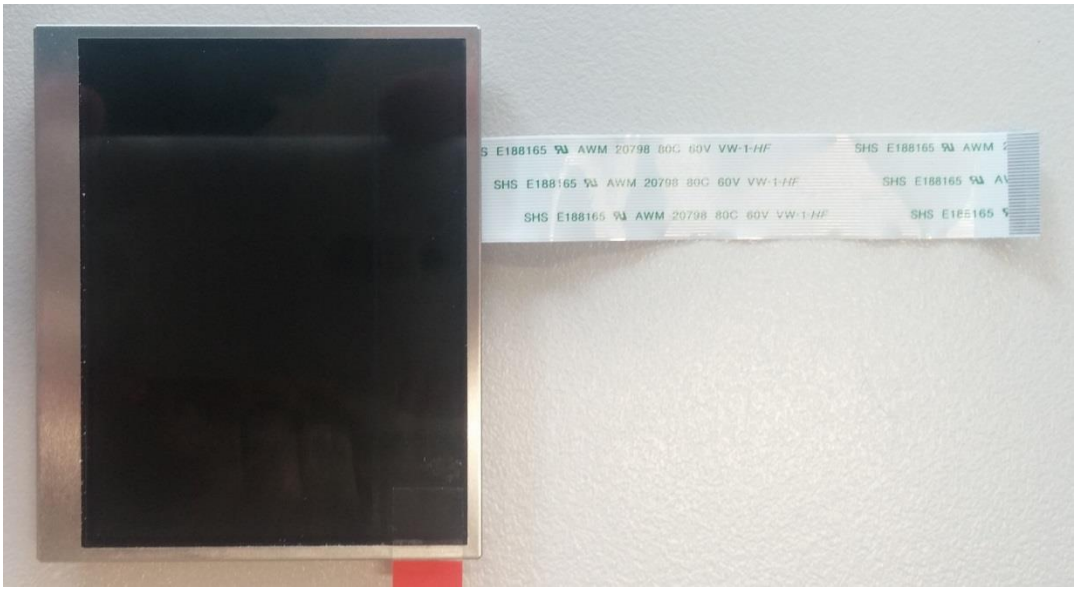

Abbildung 25 Display mit Flachbandkabel

Hauptsächlich wurde von uns mit dem Display Text dargestellt, die Zeichnung am unteren Rand in den Modi 1 und 3 stellt dabei die Ausnahme dar. Bei der Darstellung von Text lassen sich Strings durch eine Bibliotheksfunktion auf dem Display ausgeben, dabei lässt sich die Startposition auf dem Display vorgeben, sowie die Farbe des Textes und des Hintergrunds.
Für das Zeichnen gibt es verschiedene Funktionen. Die von uns am häufigsten verwendete Funktion war Draw_Line(), mit der man eine Linie zeichnen lassen kann. Um die Darstellung der Übertragung zu zeichnen, wird das letzte Zeichen in Baudot-Codierung in seine Bits zerlegt und jedes Bit in ein

Array geschrieben. Dann wird jedes Bit entweder oben oder unten gezeichnet, je nachdem ob es eine 1 oder eine 0 ist. Wenn zwei aufeinanderfolgende Bits unterschiedlich sind, wird zusätzlich noch eine Linie von oben nach unten gezeichnet, um den Übergang zu kennzeichnen. Der noch zusätzlich dargestellte Ruhepegel, das Stoppbit und das Startbit werden jedes Mal auf dieselbe Weise gezeichnet. Durch Farben wurden die einzelnen Bereiche noch voneinander abgetrennt.
Das Ergebnis des Ganzen ist in Abbildung 26 dargestellt.

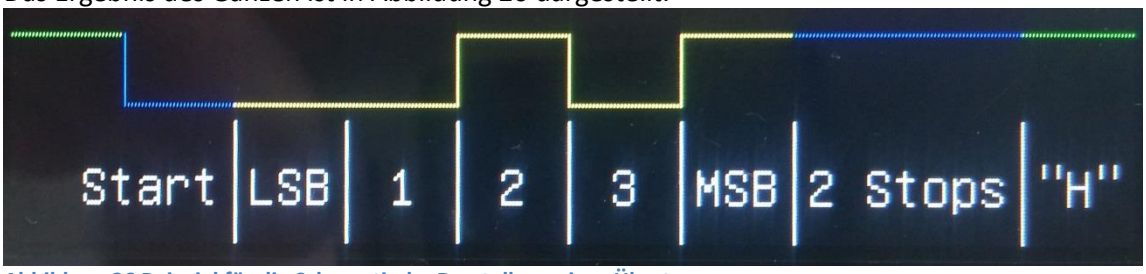

Abbildung 26 Beispiel für die Schematische Darstellung einer Übertragung

7.7 Drucker

Um die Kommunikation mit dem Fernschreiber auf Papier mit zu protokollieren, wurde ein Thermodrucker mit der Bezeichnung „BT-2“ verbaut. Dieser ist zum Beispiel bei Adafruit verfügbar[14] und verwendet eine serielle Schnittstelle mit 3,3 V oder 5 V Spannung und 19200 Baud. Die Versorgungsspannung liegt zwischen 5 V und 9 V [9], bei uns bot es sich an, 5 V zu verwenden.

Abbildung 27 Der verwendete Thermodrucker

In Abbildung 27 ist der Thermodrucker zu sehen, der in unser Gehäuse mit eingebaut wurde. Im Vordergrund rechts sind auch vorgenommenen Modifikationen zu sehen. Da die Serielle Schnittstelle des Raspberry Pi durch den Fernschreiber belegt ist, war es nötig einen USB-zu-Seriell Adapter zu verwenden. Da in dem Gehäuse des Thermodruckers noch genügend Platz war, wurde dieser Adapter direkt an die Platine im Inneren des Thermodruckers angelötet. Ein Zugangsloch wurde

[14] https://www.adafruit.com/products/597

geschaffen und der Adapter wurde mit doppelseitigem Klebeband befestigt. Ein Anschluss für die 5 V Versorgungsspannung wurde ebenfalls von uns angebracht.
Die vom Hersteller vorgesehenen Stecker, in Abbildung 28 zu sehen, wurden von uns allerdings nicht beeinflusst, so dass der Drucker noch normal verwendet werden kann.
Abbildung 29 zeigt das Ergebnis der Modifikationen im Inneren des Thermodruckers. Die Leiter, welche unter die Platine des Thermodruckers geführt werden, sind einfach an die Unterseite der Steckverbinder angelötet. Die Beine an der Buchse für 5 V sind ein wenig zur Seite gebogen, um die Befestigung im Gehäuse zu ermöglichen.

Abbildung 28 Thermodrucker von unten

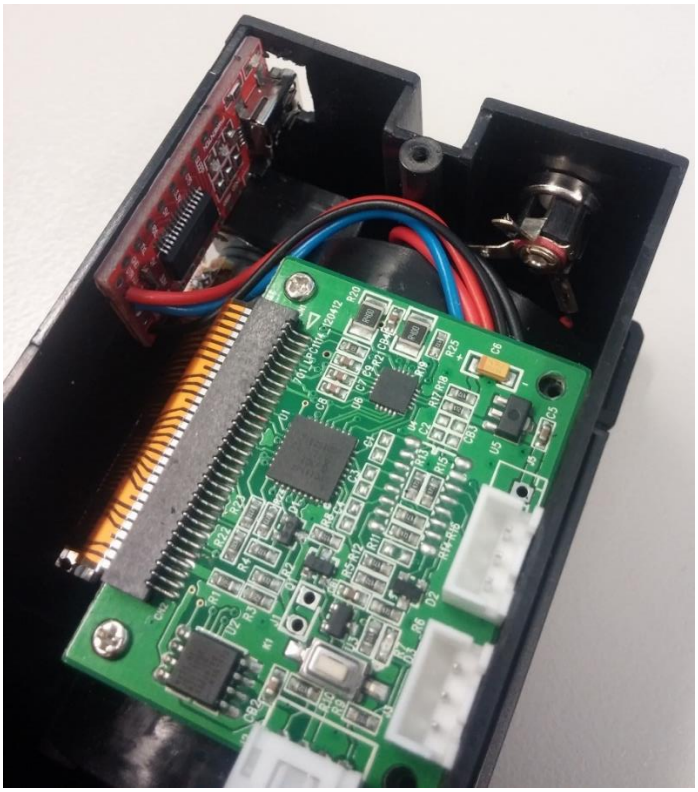

Abbildung 29 Das Innere des Thermodruckers

Die Ansteuerung des Druckers in der Software war nicht wesentlich anders als die Ansteuerung des UART und brachte dann im Endeffekt durch unsere Programmierung das Ergebnis wie in Abbildung 30 zu sehen ist.

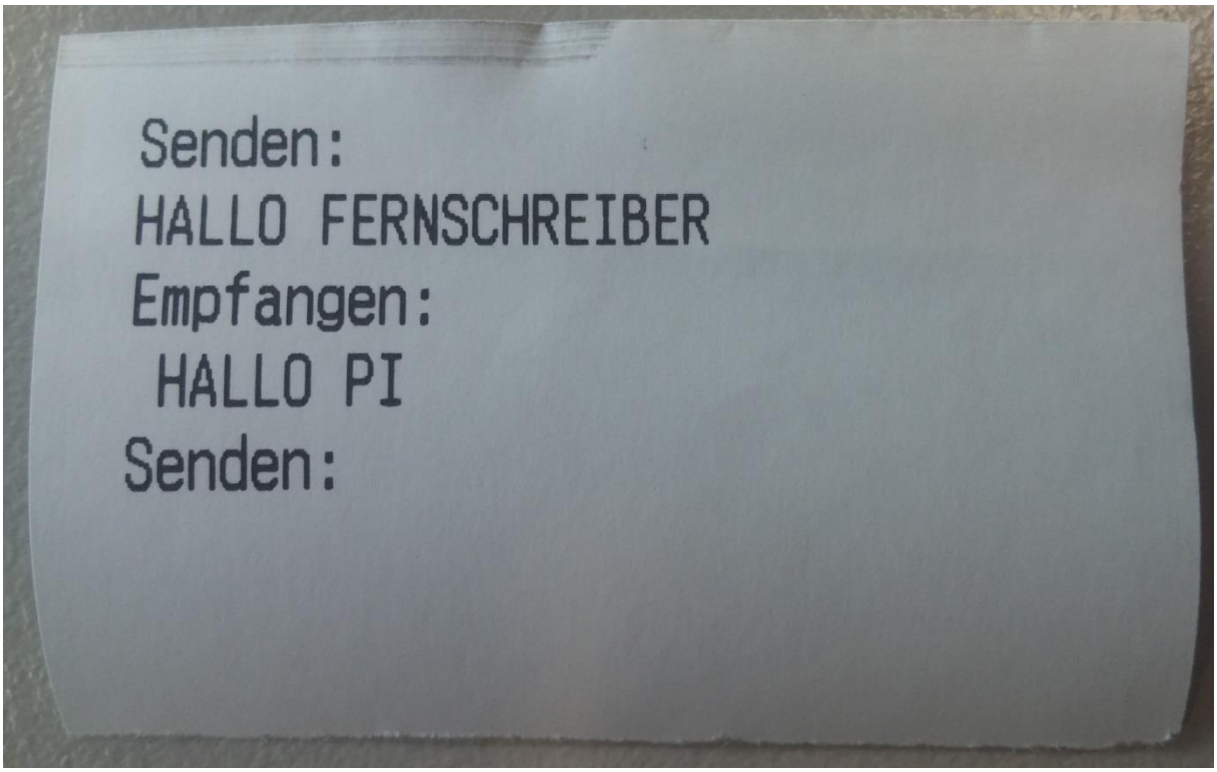

Abbildung 30 Ausdruck des Thermodruckers

8 Die Hardware

Da der Raspberry Pi die serielle Kommunikation mit 3,3 V umsetzt, der Fernschreiber aber mit einer 40 mA Stromschleife arbeitet, ist Hardware von Nöten, die diese Pegel ineinander umsetzt. Dazu ist es notwendig, sowohl das Senden, als auch das Empfangen zu beachten, was beim Fernschreiber eins ist, beim Raspberry Pi aber auf 2 Pins aufgeteilt ist. Zusätzlich wurde noch eine galvanische Trennung und weitere Schutzmaßnamen zwischen dem RPi und dem Fernschreiber umgesetzt, um den RPi vor Spannungsspitzen zu schützen.

Um das Hardware-Interface so einfach wie möglich zu halten, wurde eine Platine entwickelt, welche einfach auf den Raspberry Pi gesteckt werden kann. Um das Display trotzdem verwenden zu können, welches ebenfalls mit einer Aufsteckplatine ausgestattet ist, wurde eine Buchsenleiste mit langen Kontaktstiften gewählt, welche es ermöglicht, auf die von uns gebaute Platine die Platine des Displays ebenfalls aufzustecken.

Da der Raspberry Pi 5 V Betriebsspannung benötigt, für den Fernschreiber aber mehr Spannung benötigt wird, wurden der Einfachheit halber 2 Spannungsquellen verbaut, eine mit 5 V für den Raspberry Pi und eine mit 36 V für den Fernschreiber.

8.1 Schaltungen auf der Aufsteckplatine

Da der Tx-Pin des Raspberry Pi's eine Ausgangsspannung von 3.3V und einen maximalen Strom von 16 mA pro Pin hat, ist es nicht möglich mit dem Optokoppler direkt die 40 mA zu schalten. Bei einem Strom von 10 mA (I_F) ist der maximal zu schaltende Strom laut Datenblatt etwa 18 mA. Daher haben wir uns für einen Stromspiegel entschieden, der einen Referenzstrom von 10 mA und einen ausgangsseitigen Strom von 40 mA bereitstellt.

Um den Strom für den RX-Pin zu messen, wurde über einen Widerstand Spannung für den Optokoppler abgegriffen, auf dessen anderer Seite ein Netzwerk mit Tiefpassfilter liegt, um das Signal ein wenig zu glätten.

Da die gesamte Schaltung mit E12-Widerständen umgesetzt wurde, weil diese vorrätig waren, stimmen die Ströme und Spannungen nicht genau mit den Werten überein, für die sie dimensioniert wurden. Zum Beispiel ist der Strom im Signalpfad nicht 40 mA, sondern 41,7 mA, diese geringfügige Abweichung hat aber keinen nennenswerten Einfluss.

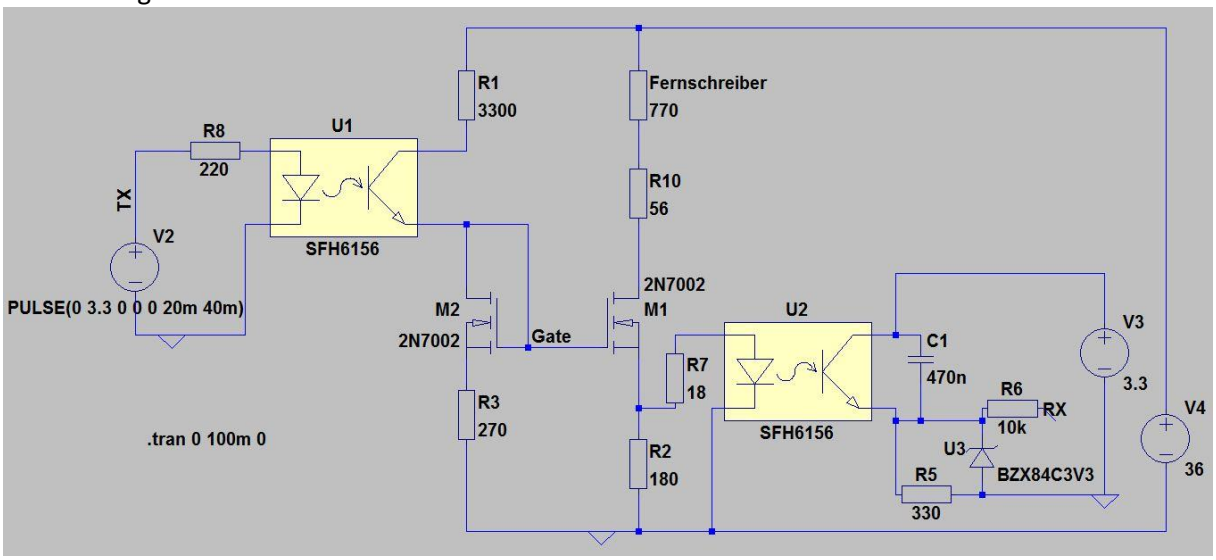

Abbildung 31 Schaltung der eigenen Platine für das Signal

Der Fernschreiber wurde in der in Abbildung 31 dargestellten Schaltung durch einen Widerstand von 770 Ω modelliert, welcher dem gemessenen Innenwiderstand des Signalpfades des Fernschreibers entspricht.

Das Signal für RX wird über den Optokoppler U2 galvanisch getrennt bereitgestellt. C1 dient dazu, das Signal zu filtern. Durch R5 fließen im Betrieb 10 mA, dadurch bildet sich oberhalb der Z-Diode U3 eine Spannung von 3,3 V. Falls Überspannungen an U3 durch Einkopplungen oder sonstige Störungen vorkommen, leitet die Z-Diode Spannungen oberhalb von etwa 3,3 V gegen Masse, und der Vorwiderstand R6 beschränkt den Strom, der in den RPi fließen könnte, um diesen zu schützen.
V2 stellt den TX-Pin des Raspberry Pi dar, RX den RX-Pin.

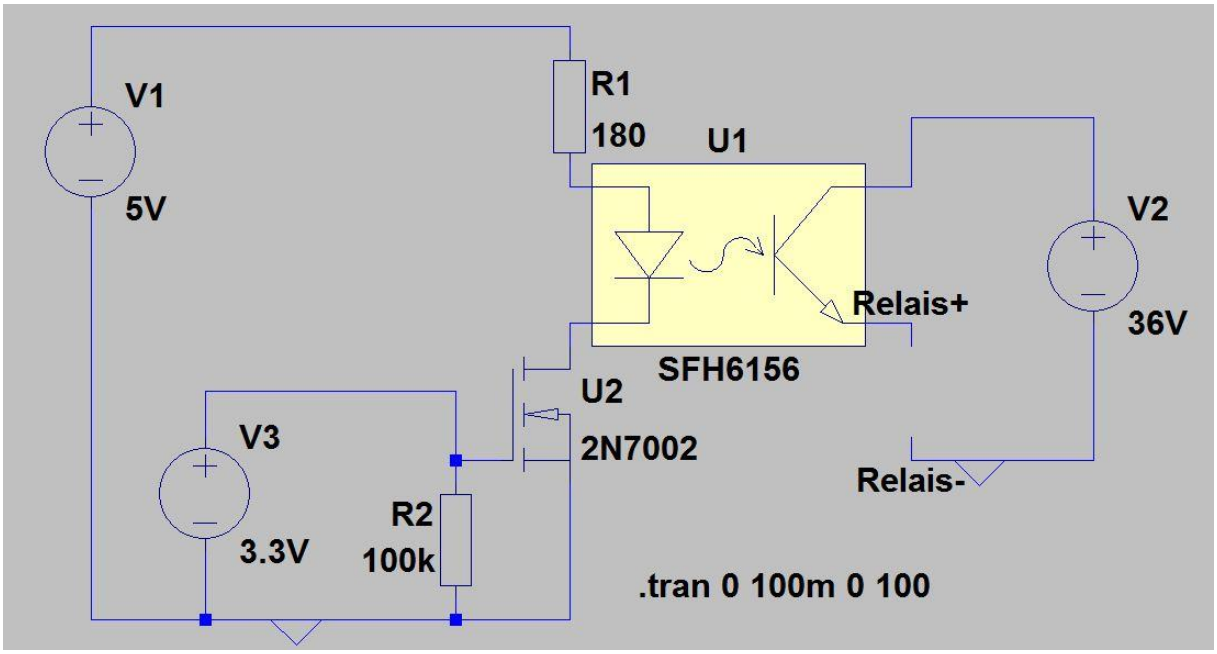

Abbildung 32 Schaltung zur Ansteuerung des Relais

Da auch ein Relais verbaut wurde, um den Fernschreiber per Software an- und abzuschalten, musste dieses auch angesteuert werden können. Die Ansteuerung ist in Abbildung 32 dargestellt. Da das Relais laut Datenblatt ca. 27 mA benötigt um zu schalten, musste auf der Diodenseite des Optokopplers ebenfalls mehr Strom fließen. Um den GPIO-Pin des Raspberry Pi nicht zu überlasten wurde deshalb die Spannung von der 5 V Quelle des RPi benutzt und mit dem GPIO-Pin ein MOSFET geschaltet. R1 dient dabei dazu den Strom durch die Diode zu begrenzen, wodurch ca. 20,5 mA durch die Diode fließen. V3 stellt den GPIO-Pin dar, der benutzt wird, um das Relais zu schalten. Der Pulldown-Widerstand R2 wurde verbaut, damit im Grundzustand ein definierter Pegel am Gate von U2 liegt.
Auf der anderen Seite des Optokopplers begrenzt das Relais selbst den Strom, wodurch kein Vorwiderstand benötigt wird.
Da das Relais zum Schalten eine Spannung von 24V und ein Strom von 27mA benötigt, muss der Optokoppler diesen Strom durchlassen. Bei diesem Strom fällt am Optokoppler eine Spannung von 12V ab, wodurch kein Vorwiderstand mehr benötigt wird.

Durch eigene Erfahrungen aus Praktika, der Ausbildung und Simulationen in LTSpice war die Entwicklung der Schaltungen und Dimensionierung der Bauteile mithilfe der Datenblätter machbar.

8.2 Layout

Das Layout wurde in Eagle erstellt. Um zu erreichen, dass unsere Platine auf den Raspberry Pi passt, wurde eine Bibliothek für den Raspberry Pi 2B verwendet, welcher von den äußeren Dimensionen dem verwendeten Raspberry Pi gleicht [10].

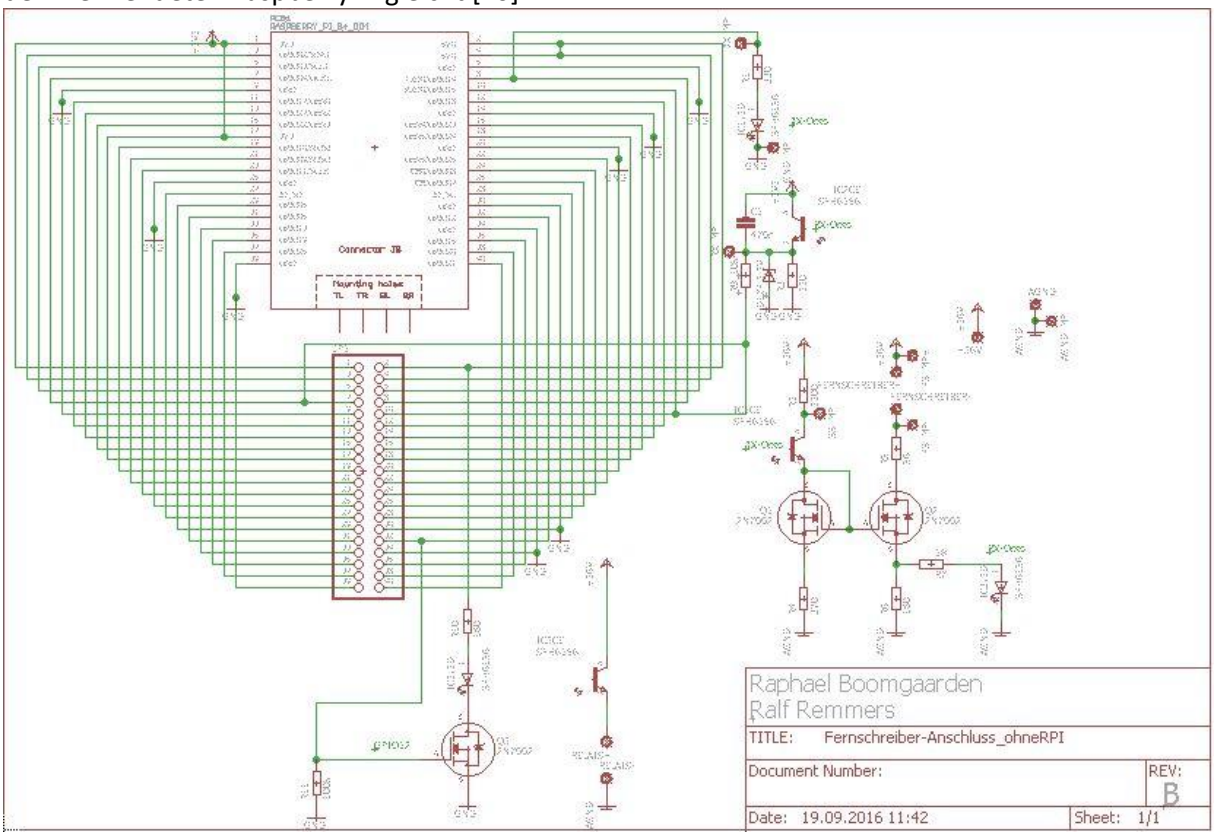

Abbildung 33 Schaltung in Eagle

Zunächst wurde die Schaltung in Eagle übertragen, wie in Abbildung 33 dargestellt ist. Im Anhang ist dasselbe Bild noch einmal größer dargestellt (Abbildung 67). Besonderheiten gegenüber den vorigen Darstellungen sind der Pinheader JP1 und eingefügte Messpunkte (gekennzeichnet durch …_MP). Um den Fernschreiber und das Relais, sowie die 36 V Spannungsquelle anzuschließen, wurden kontaktierte Löcher vorgesehen und dort Leitungen angelötet. Außerdem werden für die mit 5 V betriebene Seite und die mit 36 V betriebene Seite verschiedene Massen verwendet, um die galvanische Trennung zu gewährleisten.

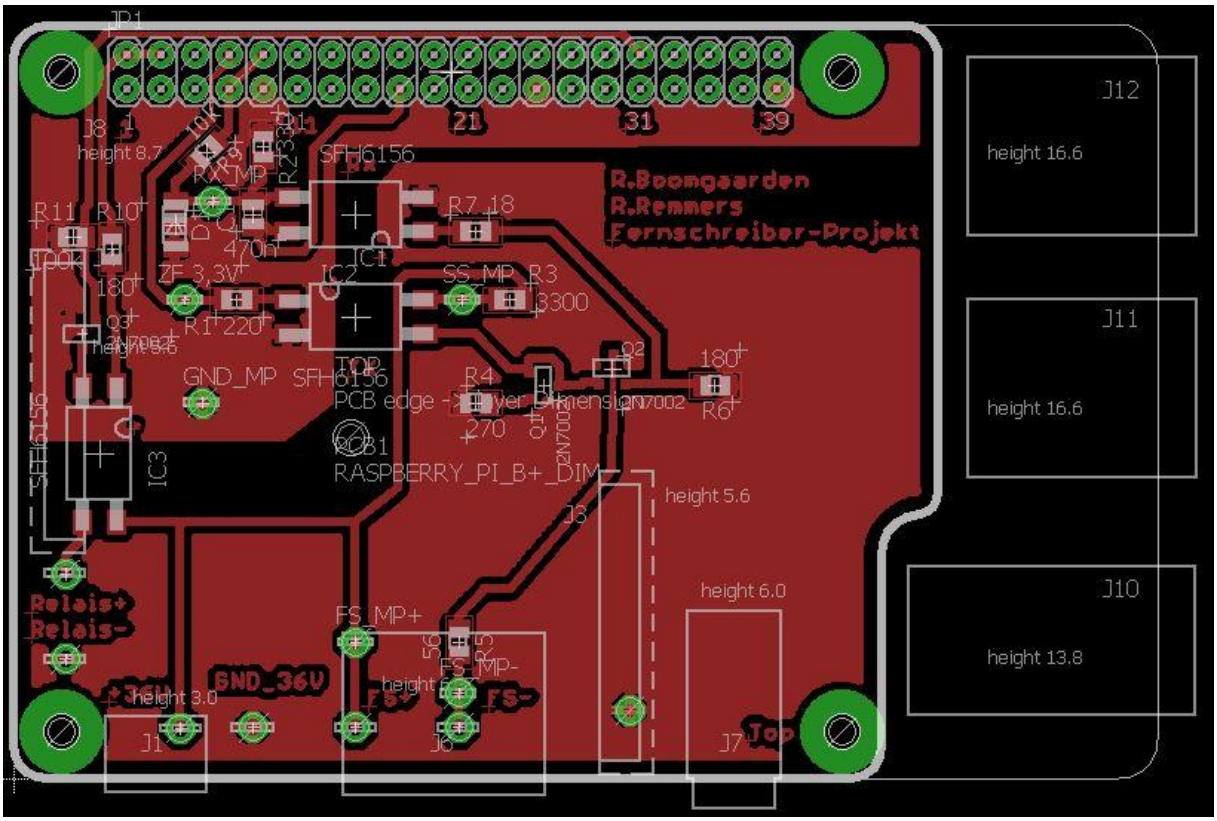

Abbildung 34 Layout mit Vorlage

In Abbildung 34 ist das Layout mitsamt der Vorlage dargestellt. Von besonderem Interesse dabei waren die Anschlüsse rechts (von oben nach unten: USB, USB und Ethernet), welche so hoch sind, dass sie nicht mehr unter unsere Platine passen würden. Sämtliche Anschlüsse links davon sind flacher, so dass unsere Platine über sie passt. Anschlüsse, die von oben verwendet werden, werden dabei von unserer Platine verdeckt und sind dadurch schwerer nutzbar.
Die Anschlüsse J1, J6 und J7 (USB-Spannungsversorgung, HDMI und Composite Video/3,5 mm Klinke) bleiben von unserer Platine unbeeinflusst, da sie von der Seite gesteckt werden. Es ist zu erkennen, dass die Bohrlöcher unserer Platine und der Platine vom Raspberry Pi genau übereinanderliegen und auch, dass die Pins im oberen Teil des Layouts genau übereinander passen. Dies war erforderlich um unsere Platine auf den Raspberry Pi stecken zu können und die Verbindung der beiden Platinen durch Schrauben zu ermöglichen.
Die Kontaktflächen um die Bohrlöcher kommen von der Raspberry Pi Vorlage und wurden für unser Layout ignoriert, da auf unserer Platine Kunststoff-Abstandshalter verwendet wurden, die eine Isolierung gewährleisten.

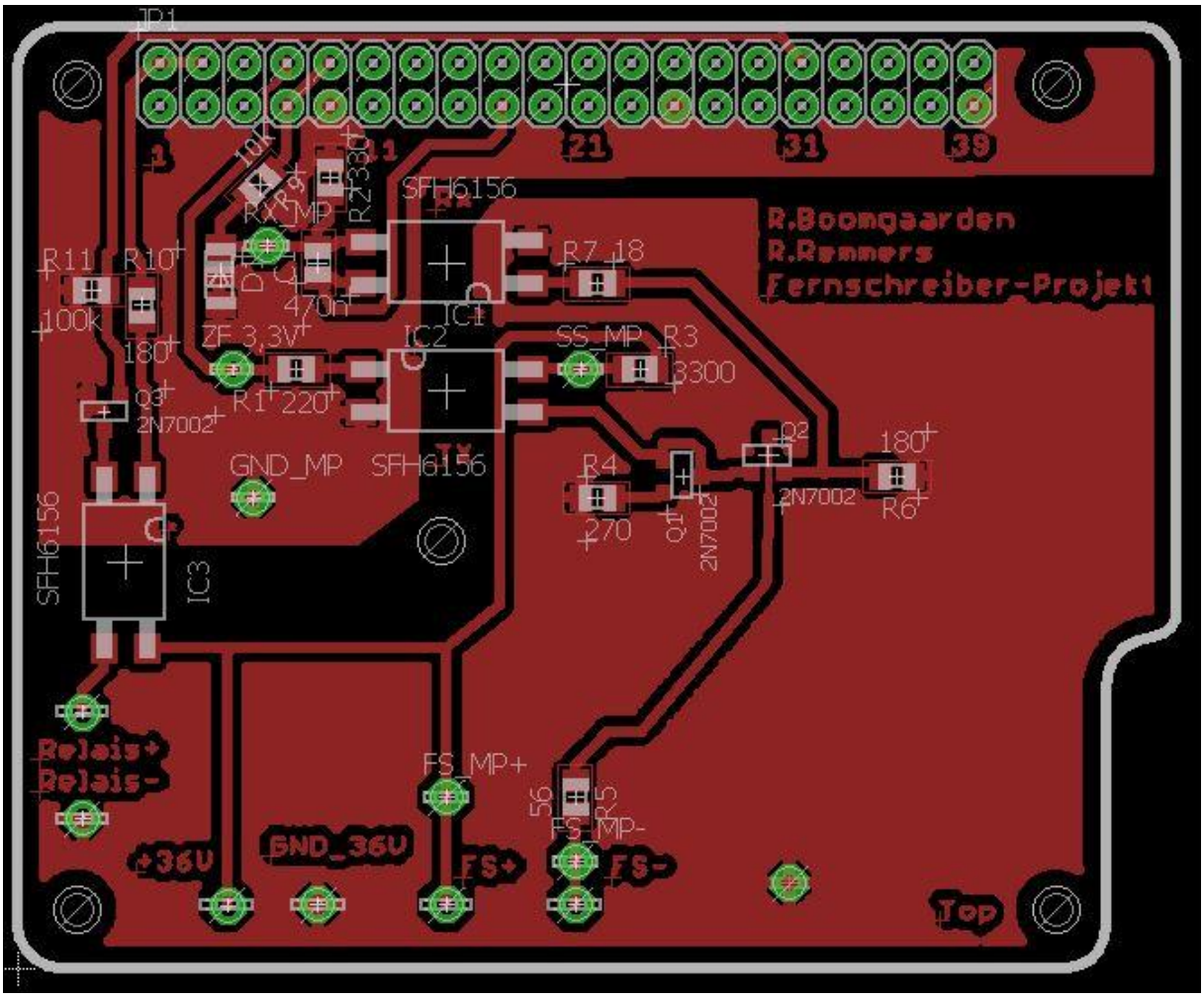

Abbildung 35 Layout ohne Vorlage

In Abbildung 35 ist das letztendlich gefräste Layout zu sehen. Es wurde gefräst, da hierdurch die Bohrlöcher nach dem Herstellungsprozess schon vorliegen und nicht per Hand gebohrt werden müssen, was eine optimale Genauigkeit gewährleistet. Außerdem hat die Platine eine nicht-rechteckige Kontur, was durch Fräsen leichter herzustellen ist.
Das Bohrloch in der Mitte dient dazu einen Abstandshalter zu befestigen, der die Platine des Displays abstützt, da diese sonst nur auf den GPIO-Pins befestigt ist.
Zwischen den zwei Masseflächen wurde ein größerer Isolationsabstand gewählt, um die Isolation zu gewährleisten.

8.3 Relais

Da ein Relais möglicherweise starke EMV-Probleme verursachen kann und damit auf unserer Platine auf dem Raspberry Pi keine 230 V anliegen, wurde eine separate Platine für das Relais entworfen und gefertigt. Diese zusätzliche Platine wurde über 2 Leitungen angebunden, welche mit einem Stecker versehen wurden, um den Ein- und Ausbau zu vereinfachen.

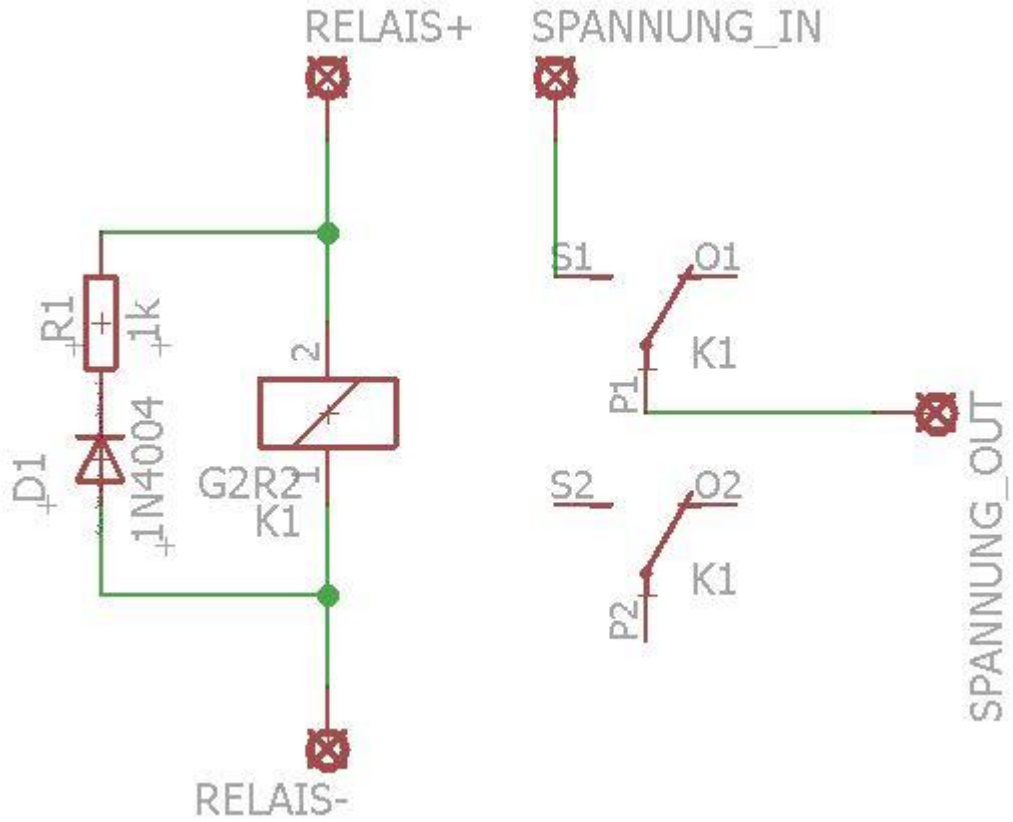

Abbildung 36 Schaltplan für die Relaisplatine

In Abbildung 36 sieht man, wie die Schaltung auf der separaten Platine aufgebaut ist. Die Bauteile RELAIS+, RELAIS-, SPANNUNG_IN und SPANNUNG_OUT sind Bohrungen mit Kontaktflächen, in die Leitungen eingelötet wurden. Ein Leitungspaar wurde für die Versorgung der Schaltspule angebracht und das andere für den zu schaltenden L-Leiter der 230 V des Fernschreibers. Leider hatte das von uns verwendete Relais zwar für Normal Offen, Normal Geschlossen und den Schalter jeweils 2 Kontakte, diese waren für die jeweiligen Kontakte aber intern gebrückt. Dadurch ließ sich dann nur ein Leiter für 230 V schalten.
Damit im Grundzustand nicht an 4 Polen 230 V anliegen, wurde der normal offene Kontakt als Eingang gewählt und der Schalter als Ausgang.
Um die restliche Schaltung zu schützen, wurde noch eine Freilaufdiode verbaut mit einem Vorwiderstand, um das Entladen der Spule zu beschleunigen.

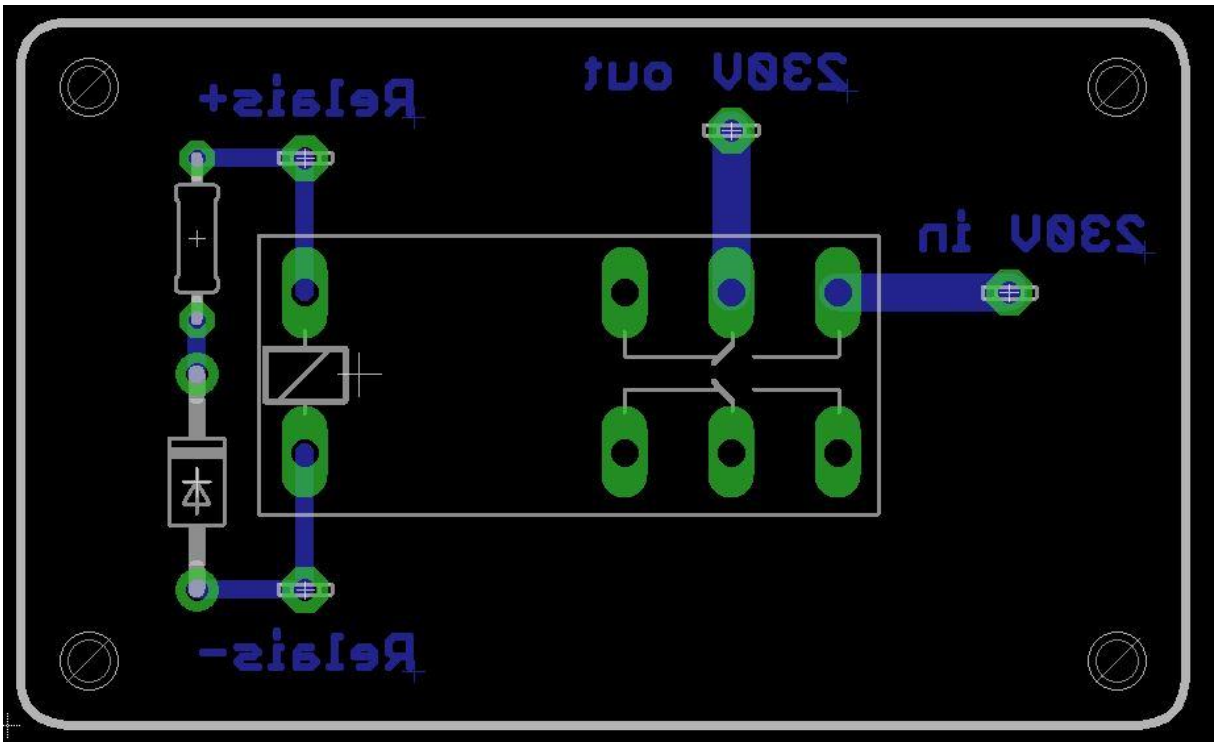

Abbildung 37 Platinenlayout der Relaisplatine

Abbildung 37 stellt das Layout der Relaisplatine dar. Aufgrund der Einfachheit der Schaltung kommt dieses leicht mit nur einer Lage aus, welche die untere Lage ist. Deshalb ist die Schrift auch invertiert. Mit den vier Bohrlöchern wurde die Platine dann am Gehäuse befestigt.

9 Das Gehäuse

Um den Raspberry Pi zusammen mit den zwei Netzteilen, der separaten Relaisplatine, dem Drucker und den Anschlüssen für Netzspannung unter zu bringen, wurde aus Aluminium ein Gehäuse gebaut. Dieses dient auch gleichzeitig der Abschirmung gegenüber elektromagnetischen Strahlen vom Fernschreiber, da der Fernschreiber davon einiges abstrahlt. Konstruktiv gesehen besteht das Gehäuse aus einem Boden und einem Deckel, die durch Aluminium-Profile miteinander verbunden sind. In die Aluminium-Profile sind die Seitenteile eingesteckt. Sämtliche Seiten, Boden und Deckel sind geerdet und auf dem Boden ist eine Folie aufgeklebt, um die Isolierung von Platinen zum Boden zu gewährleisten. Zwischen den Kaltgerätesteckern auf der Rückseite wurde noch ein weiteres Profil befestigt, um die Rückwand beim Einstecken der Buchsen abzustützen.
Die Platinen sind mit Abstandshaltern und Schrauben am Boden befestigt. In der rechten Seitenwand sind Buchsen für USB und den Fernschreiber eingelassen. In der hinteren Seitenwand sind die Anschlüsse für 230 V, in Abbildung 38 oben zu sehen. Von links nach rechts sind dort Anschlüsse für 230 V Normalnetz, 230 V mit Trenntransformator-Eingang und 230 V mit Trenntransformator-Ausgang. Diese sind als Kaltgerätestecker bzw. Kaltgerätebuchse für den Ausgang ausgeführt.
An dem 230 V Normalnetz sind die 36 V (oben) und 5 V (unten) Netzteile angeschlossen.
Vom Kaltgerätestecker für 230 V mit Trenntransformator geht der N-Leiter direkt zur Kaltgerätebuchse, von welcher der Fernschreiber betrieben wird. Der L1-Leiter wird über das Relais in der hinteren rechten Ecke geschaltet und dann ebenfalls an die Kaltgerätebuchse angeschlossen. Die Schutzerde der beiden mit Trenntransformator betriebenen Buchsen ist verbunden, damit durch die freien Anschlüsse keine Verwirrung entsteht.
Das 36 V Netzteil ist auf der Niederspannungsseite mit zwei Leitungen mit unserer Platine verbunden. Das 5 V Netzteil ist auf der Niederspannungsseite über einen Micro-USB Stecker mit dem Raspberry Pi verbunden. Es wäre möglich gewesen, die Versorgung über die GPIO-Pins anzuschließen, aber beim RPi ist am Micro-USB Eingang eine Schutzschaltung verbaut, dadurch ist ein wenig mehr Schutz für den RPi gewährleistet. Ebenfalls an dem 5 V Netzteil ist die Spannungsversorgung für den Thermodrucker verbaut. Da dieser bis zu 1,5 A beim Drucken benötigen kann, haben wir ihn direkt mit dem Netzteil verbunden.
Das USB-Kabel für den Thermodrucker ist mit einem USB-Anschluss des Raspberry Pi verbunden, ebenso wie die nach außen geführte USB-Buchse. Mit unserer Platine sind ebenfalls die Anschlüsse an den Fernschreiber über eine Steckverbindung verbunden. Die Anschlüsse sind als Bananenbuchsen in die rechte Seitenwand eingelassen. Als drittes ist das Relais mit unserer Platine verbunden, ebenfalls über einen Steckverbinder. Die Steckverbinder wurden verwendet, um den Ein- und Ausbau der verschiedenen Bauteile zu vereinfachen.
Ganz vorne im Gehäuse (auf der Abbildung 38 ganz unten) ist noch ein Piezolautsprecher zu sehen, der eingebaut wurde, um dem Raspberry Pi zu ermöglichen, ebenfalls zu klingeln, wenn der Fernschreiber das entsprechende Zeichen sendet.

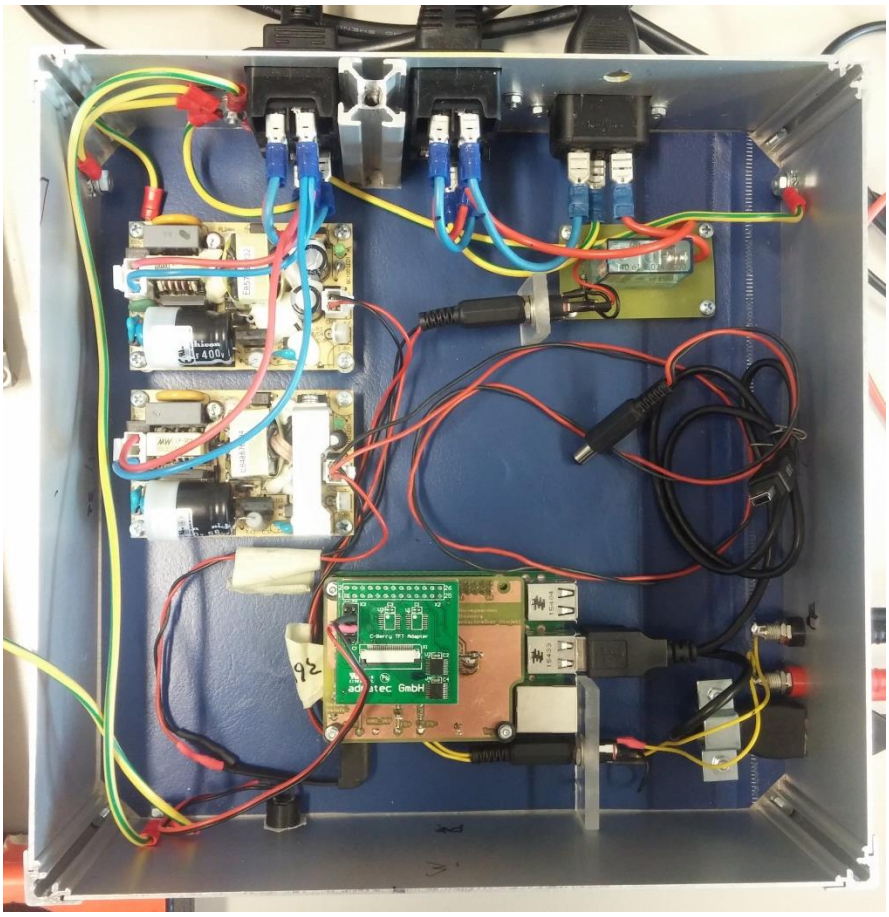

Abbildung 38 Ansicht des Gehäuses ohne Deckel

Abbildung 39 Gehäuse von hinten

In Abbildung 39 sieht man das Gehäuse von hinten. Auf der linken Seite ist gelb hinterlegt, was nur über einen Trenntransformator betrieben werden soll. Über der Kaltgerätebuchse links sieht man eine Bohrung, wo eigentlich ein Zurücksetz-Knopf hinsollte. Da dieser Knopf mit seinen Zuleitungen allerdings sehr nahe am Relais wäre, war es uns nicht möglich, ihn ausreichend zu schirmen, so dass der Raspberry Pi sich beim Einschalten des Fernschreibers oft neu gestartet hat. Daher wurde der Zurücksetz-Knopf weggelassen.
In der Abbildung sieht man ebenfalls, dass die beiden rechten Kaltgerätestecker mit Schaltern und Sicherungen ausgestattet sind. In dem Sicherungshalter für das Normalnetz befindet sich eine 6,3 A Sicherung, in dem Sicherungshalter für den Betrieb mit Trenntransformator eine 2,5 A Sicherung.

Abbildung 40 Gehäuse von der rechten Seite

Abbildung 40 zeigt die rechte Seite des Gehäuses, rechts unten sieht man die Schraube, an der von innen die Erdung befestigt ist. Links unten ist der USB-Anschluss für die Tastatur und die zwei Bananenbuchsen, um den Fernschreiber anzuschließen.

Abbildung 41 Gehäuse von oben

Auf Abbildung 41 ist das Gehäuse von oben zu sehen. Gut sichtbar sind die Schrauben, die den Deckel in den Ecken mit den Aluminium-Eckprofilen verbinden und hinten in der Mitte die Schraube, die den Deckel mit dem Profil zum Abstützen der Rückseite verbindet.

Vorne ist das Display befestigt. Auf der rechten Seite mittig ist der Drucker in das Gehäuse eingelassen. Hinten sind die verschiedenen Kaltgerätestecker noch beschriftet. Der Drucker ist von innen nicht befestigt, sondern einfach herausnehmbar. Dabei muss man allerdings die linke Seite zuerst herausheben, da rechts die Leitungen für Spannung und USB eingesteckt sind.

Von vorne sieht man das Gehäuse in Abbildung 42. Dort ist gut zu sehen, dass das Display mithilfe von abgewinkeltem Aluminiumblech schräg befestigt wurde, so dass man das Angezeigte besser sehen kann.

Abbildung 42 Gehäuse von vorne

10 Messungen

Im Verlauf des Projekts wurden verschiedene Messungen durchgeführt, die hier kurz vorgestellt werden.

Zunächst wurde das vom Fernschreiber ausgehende Signal über einen 100 Ω Widerstand gemessen (siehe Abbildung 43). Dies war stark verrauscht, daher wurde von uns ein Filter auf einer Lochrasterplatine mit bedrahteten Bauteilen angefertigt. Dies verbesserte das Signal erheblich, wie in Abbildung 44 zu sehen, aber es waren immer noch Spannungsspitzen vorhanden. Um diese noch vorhandenen Spannungsspitzen zu vermindern, wurden dann im Design der Platine eine 3,3 V Zenerdiode und ein Vorwiderstand, der den Stromfluss in den Raspberry Pi begrenzt, vorgesehen.

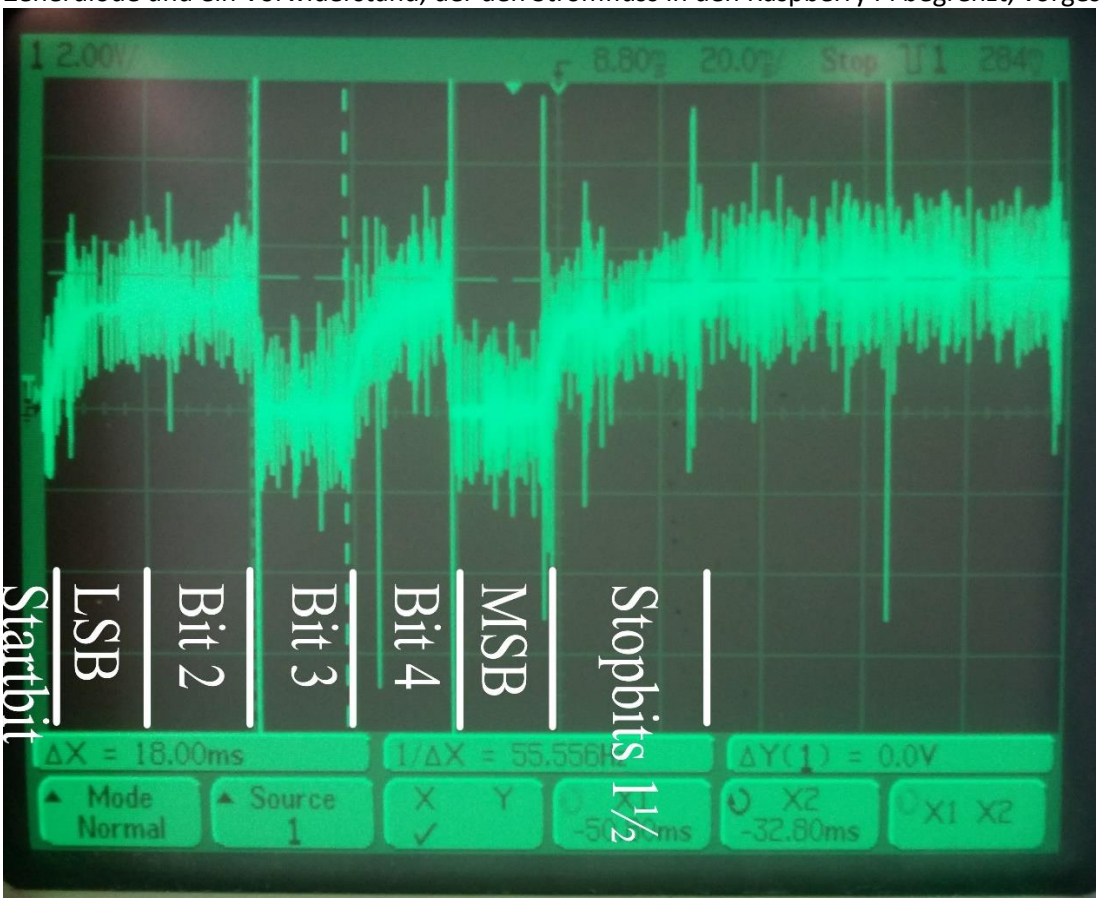

Abbildung 43 Erste Messung des Fernschreibersignals, ein S wurde übertragen

Abbildung 44 Das gefilterte Signal, ein S wurde übertragen

Das Signal des Fernschreibers ist allerdings mittlerweile nicht mehr so verrauscht, wie in Abschnitt 4.3 „Modifikationen am Fernschreiber“ beschrieben.
Das vom Raspberry Pi ausgehende Signal ist wesentlich sauberer, wie in Abbildung 45 zu sehen ist.

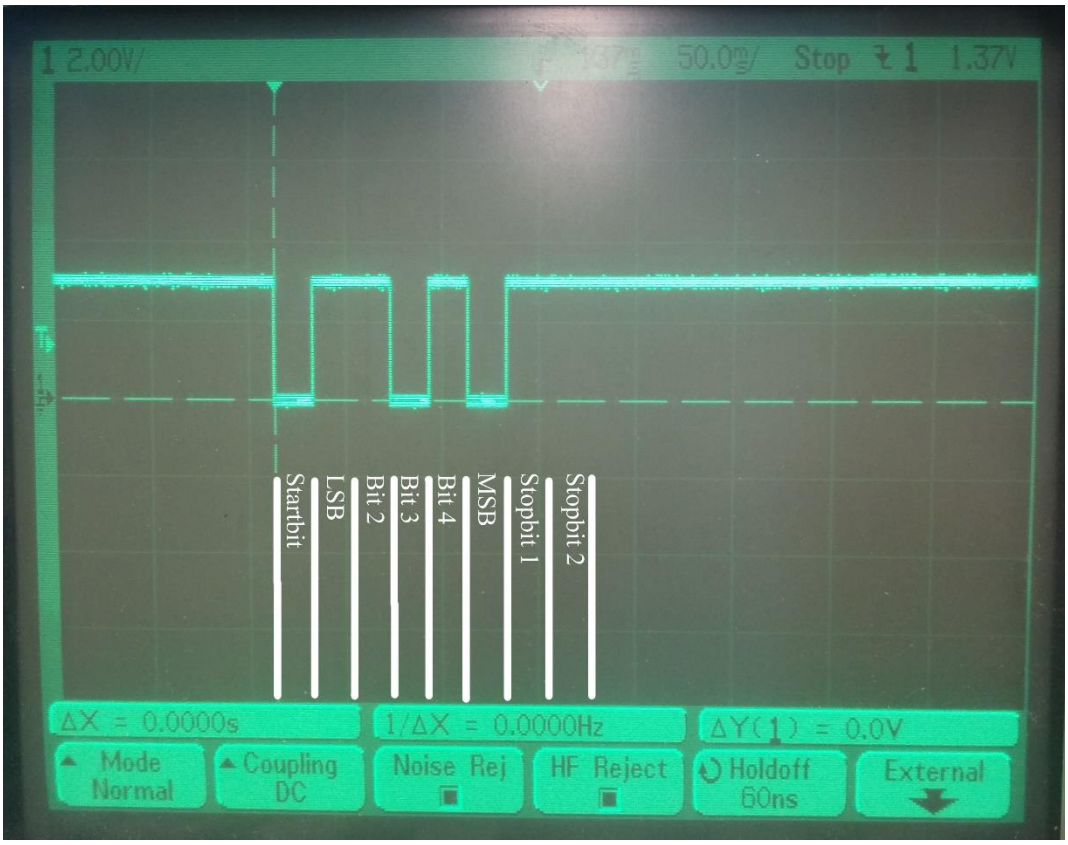

Abbildung 45 Vom Raspberry Pi ausgehendes Signal, ein S wurde übertragen

Im RX-Pfad haben wir ebenfalls gemessen, allerdings vor dem Vorwiderstand, gezeigt in Abbildung 46.

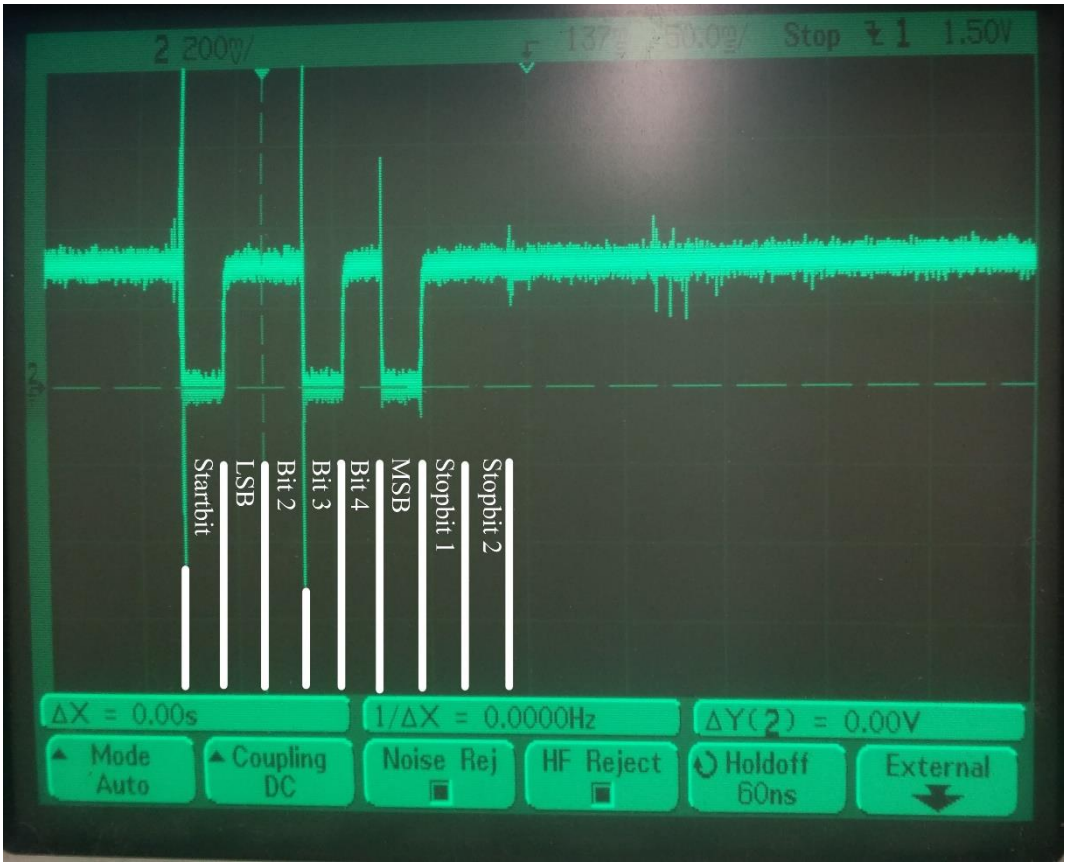

Abbildung 46 Signal im RX-Pfad mit offenem Deckel, ein S wurde übertragen

Es ist noch Rauschen zu erkennen, und man sieht Spannungsspitzen, dies ist allerdings bei geschlossenem Deckel deutlich weniger, wie Abbildung 47 zeigt. Die verbliebenen Spannungsspitzen sind für den Raspberry Pi kein Problem, da der Vorwiederstand den Strom auf wenige µA beschränkt.
Die Spannungsspitzen kommen von den im Fernschreiber verbauten Spulen, zu erkennen ist dies, da nur beim Unterbrechen des Stromes Spannungsspitzen auftreten. Bei einer Unterbrechung im Strom hat die Spule noch Energie in Form eines magnetischen Feldes gespeichert. Das magnetische Feld kann mangels Strom nicht mehr aufrechterhalten werden. Dadurch werden Ladungsträger frei, welche dann eine hohe Spannung verursachen. Der Einsatz von Freilaufdioden ist nicht möglich, da die Steilheit der Flanken zu sehr abnimmt, was eine Zeichenerkennung verhindert.
Eine Erklärung dafür, dass dies bei geschlossenem Deckel weniger auftritt, haben wir nicht.

Abbildung 47 Signal im RX-Pfad mit geschlossenem Deckel, ein S wurde übertragen

11 Noch bestehende Fehler, bekannte Bugs

Leider gibt es bei diesem Projekt, wie in fast jedem, einige kleine Probleme, die sich nicht mehr im Zeitrahmen beheben ließen. Diese sind allerdings eher qualitativer als funktioneller Natur.
Zunächst sei erwähnt, dass der Client-Raspberry Pi im Modus 3 manchmal einige Zeit braucht, um zu aktualisieren. Dies liegt daran, dass FTP zur Übertragung verwendet wurde, welches zunächst für jede Datei eine Verbindung aufbauen muss, was Zeit kostet. Es wird alle 5 Sekunden eine Übertragung per FTP angestoßen, um die vom Fernschreiber empfangenen Daten auszulesen. Während diese Übertragung läuft, wird allerdings das Benutzerinterface blockiert. Die Übertragungszeit variiert teilweise erheblich, somit ist die Blockierung des Nutzers nicht einheitlich lang. Wenn man Text an den Fernschreiber senden will, wird dieser Text sofort übertragen. Dabei kann allerdings das Eintippen des Textes von der anderen Übertragung unterbrochen werden.
Tippt man den Text genau zwischen zwei Übertragungen ein, geht dies relativ schnell.
Ein weiterer Punkt ist, dass Fernschreiber und Raspberry Pi sich zunächst auf Buchstaben-, oder Zahlenmodus durch das Übertragen eines Steuerzeichens einigen müssen, sonst ist der Anfangsmodus von Raspberry Pi und Fernschreiber eventuell nicht identisch. Durch das Festlegen von Buchstaben- oder Zahlenmodus ist sichergestellt, dass beide Geräte sich im selben Modus befinden.
Hat man den Anfangsmodus nicht definiert, empfängt der Raspberry Pi das erste von ihm gesendete Zeichen, da das vom Tx-Pin ausgesendete Signal auch immer am Rx-Pin anliegt. Es kann auch vorkommen, dass übertragene Zeichen falsch interpretiert werden, da alle Bitfolgen doppelt belegt sind.
Die Festlegung des Buchstaben- oder Zahlenmodus zu Beginn einer Übertragung ist allerdings auch zwischen zwei Fernschreibern nötig, damit sich beide im selben Modus befinden.
Als letztes muss erwähnt werden, dass man mit dem Fernschreiber nur vergleichsweise langsam schreiben kann, sonst kommen falsche Zeichen beim RPi an. Woran dies liegt, konnte nicht schlussendlich festgestellt werden, unsere Vermutung ist aber, dass es daran liegt, dass der Fernschreiber 1,5 Stoppbits benutzt, der Raspberry Pi aber 2 erwartet. Da der UART vom Raspberry Pi allerdings keine 1,5 Stoppbits verwenden kann, ist dieses Problem nicht ohne größeren Aufwand zu lösen.

12 Fazit

Schlussendlich lässt sich sagen, dass das Projekt aufgrund der Verbindung diverser bisher im Studium gelernter Dinge im Bereich der Hard- und Software eine lehrreiche Erfahrung war.
Zwischenzeitlich traten verschiedenste Herausforderungen auf, zu welchen wir dann gemeinsam Lösungen fanden.
Besonders interessant war für uns die Kombination der sehr alten und sehr neuen Technologien und das die serielle Kommunikation noch ähnlich funktioniert wie vor fast 100 Jahren.
Besonders geschätzt haben wir das freie Arbeiten, bei dem wir selbst Lösungen erarbeiten mussten und es keine Schema-F-Lösung gab. Dies wird sicherlich auch in Hinsicht auf das Berufsleben eine nützliche Erfahrung sein.
Alles in allem sind wir froh darüber, dieses Projekt gemacht zu haben und schätzen die gewonnenen Erfahrungen und Kenntnisse.

Anhang

A Anleitung für die Inbetriebnahme

A.1 Anschlüsse

Um den Fernschreiber mit der Raspberry Pi-Box kommunizieren zu lassen, sollten erst beide Netzschalter auf der Rückseite (Abbildung 48) ausgeschaltet werden. Für die „Normalnetz“ – Buchse kann jede Steckdose verwendet werden. Die Spannungsversorgung des Fernschreibers (gelb) ist aus Sicherheitsgründen über einen Trenntransformator mit 230 V, welcher mindestens 3,8 A zur Verfügung stellen kann, zu realisieren. Da in den Laboren genügend Trenntransformatoren bereit stehen umfasst dieses Projekt keinen.

Abbildung 48 Rückseite Servergehäuse

An der rechten Seite der Box (Abbildung 49) wird die Stromschleife zwischen Fernschreiber (Abbildung 50) und Raspberry über die Bananenbuchsen angeschlossen sowie eine USB-Tastatur.

Abbildung 49 Rechte Seite Servergehäuse

Abbildung 50 Anschlussplatine Fernschreiber

Jetzt können beide Netzschalter an der Raspberry Pi-Box eingeschaltet werden, nach einiger Zeit erscheint dann das Auswahlmenü auf dem Display.
Wenn alles richtig angeschlossen und eingeschaltet ist, sollte das Modus Auswahlmenü wie in Abbildung 51 erscheinen. Sieht das Menü aus wie in Abbildung 52, hat der Raspberry den Fernschreiber nicht erkannt.

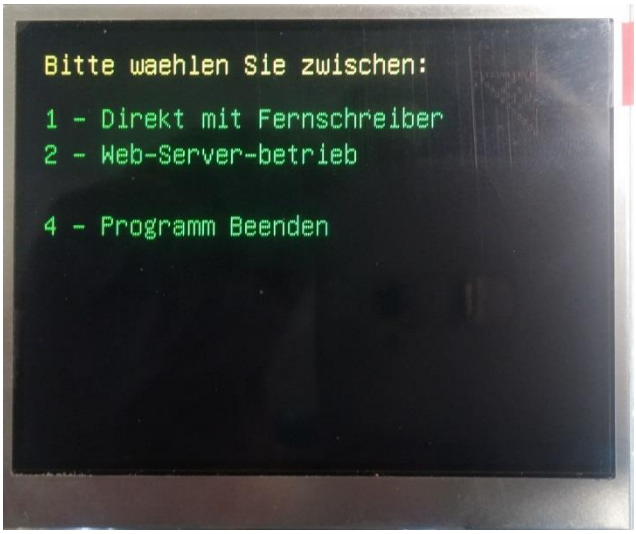

Abbildung 51 Auswahlmenü Server

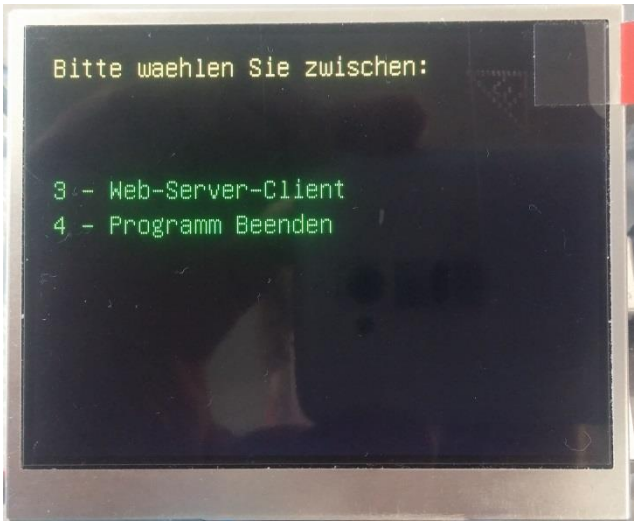

Abbildung 52 Auswahlmenü Client

Sollte der Fernschreiber nicht erkannt werden, beenden Sie mit „4“ das Programm, hiermit wird der Raspberry heruntergefahren. Schalten Sie die Netzschalter wieder aus und überprüfen Sie die Leitungen nochmals, danach wieder einschalten und erneut starten.
Sie haben die Auswahl zwischen Modus 1 und Modus 2. Danach folgt die Abfrage, ob man den Thermodrucker nutzen will, wie in Abbildung 53 dargestellt.

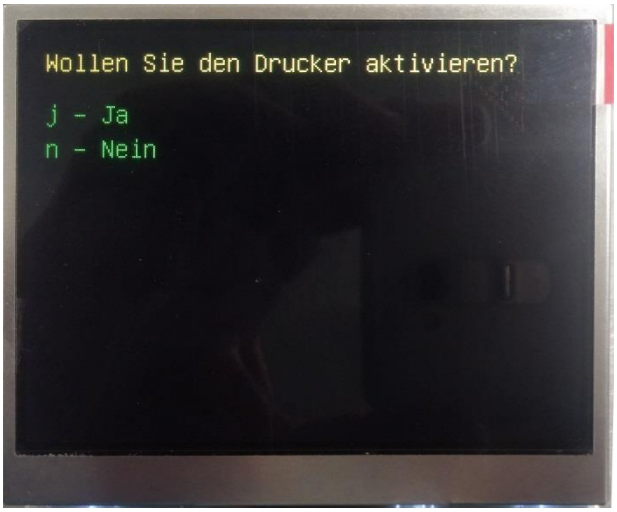

Abbildung 53 Drucker Abfrage

A.2 Modus 1

Der Modus 1 ist die direkte Kommunikation zwischen Raspberry und Fernschreiber, wie in Abbildung 54 dargestellt.

Abbildung 54 Modus 1

Um den Fernschreiber einzuschalten, gibt es den Kommandobefehl „CONNECT“ (siehe Tabelle 3). Einschalten bedeutet in diesem Zusammenhang das Schließen des 230 V-Relais, also ein Versorgen des Fernschreibers mit Netzspannung, ausschalten das Öffnen des Relais. Danach muss man entweder am Fernschreiber oder am Raspberry auswählen, ob als erstes Buchstaben oder Zahlen verschickt werden:

- Fernschreiber → Buchstabe / Ziffer Taste
- Raspberry → " Taste für Buchstabe / $ Taste Zahl

Nun kann man untereinander kommunizieren, aber immer nacheinander, nicht gleichzeitig. Der Raspberry sendet dann, wenn sich entweder 35 Zeichen in der Zeile befinden, oder die Enter/Return Taste gedrückt wird.
Der Fernschreiber hingegen sendet sofort, wobei zu beachten ist, das 50 Baud zwar heißt, dass der Fernschreiber 6 2/3 Zeichen pro Sekunde übertragen könnte, allerdings vergeht vom Tastenanschlag bis zum Senden Zeit, wodurch maximal 1 Zeichen pro Sekunde gesendet werden kann. Wenn man schneller schreibt, kommt es zu Fehlern.
Mit dem Kommandobefehl „DISCONNECT“ wird der Fernschreiber ausgeschaltet.

A.3 Modus 2

Im Web-Server-Betrieb (Abbildung 55) dient der Raspberry Pi als Server mit WLAN-Hotspot.

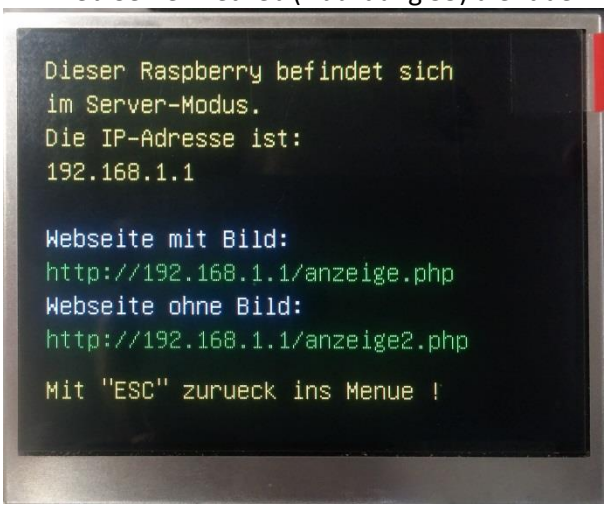

Abbildung 55 Modus 2

Die Zugangsdaten für dieses WLAN-Netzwerk lauten:

SSID	= Fernschreiber
Passwort	= Fernschreiber

Es ist gibt 2 Möglichkeiten mit dem Fernschreiber zu kommunizieren:

1. Über die PHP-Seiten anzeige.php (für PCs (Abbildung 56)) oder anzeige2.php (für Smartphones oder Tablets (Abbildung 57)), unter den auf dem Display (Abbildung 55) angezeigten Adressen.
2. Über den Raspberry-Client im Modus 3.

Abbildung 56 Anzeige.php

Abbildung 57 Anzeige2.php

A.4 PHP-Seiten

anzeige.php (Abbildung 56) und anzeige2.php (Abbildung 57) unterscheiden sich nur in zwei Punkten. Auf der anzeige.php wird unten rechts das Signal grafisch dargestellt und Anzeige2.php ist mehr für kleine Displays gedacht, da hier die Schriftgröße angepasst ist. Da beim Öffnen der Tastatur auf dem Smartphone die Anzeige des Signals verschoben wird und dadurch keine Eingabe mehr möglich ist, haben wir hier auf die Grafik verzichtet.
Die Funktion ist für den Nutzer ähnlich wie beim Modus 1 (Abbildung 54). Gesendet wird mit dem Button „Abschicken“ oder mit Enter, allerdings wird hier in eine Datei geschrieben, die in regelmäßigen Abständen ausgelesen wird, daher kommt es zu leichten Zeitverzögerungen von unterhalb einer Sekunde. Dies gilt auch beim Empfangen.

A.5 Modus 3

Dieser Modus (Abbildung 58) ist optisch identisch mit dem Modus 1, allerdings werden auch hier Dateien zur Kommunikation zwischen den beiden Raspberrys verwendet, die über FTP ausgetauscht werden.

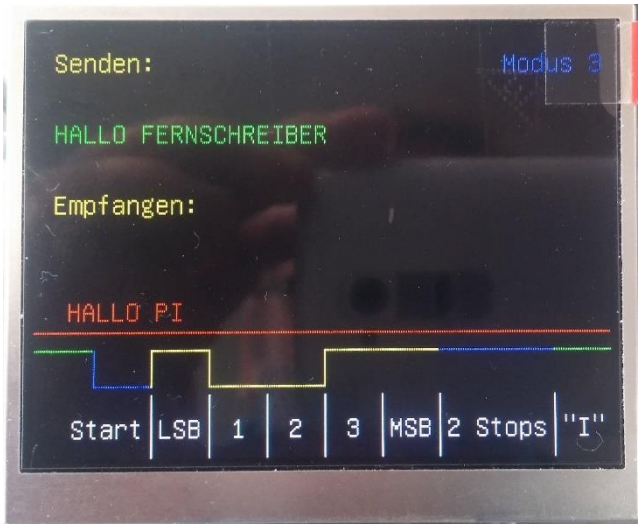

Abbildung 58 Modus 3

Der Raspberry-Client startet ebenfalls direkt in das Programm, da allerdings kein Fernschreiber angeschlossen ist, erscheint das Menü wie in Abbildung 52. Wichtig beim Starten des Clients ist es, dass der Server bereits hochgefahren ist, da sonst das Netzwerk nicht gefunden wird. Es gibt als Auswahl nur Modus 3 und Programm Beenden. Es folgt eine Abfrage der Server IP-Adresse, wie in Abbildung 59 zu sehen. Diese wird auf dem Server angezeigt (siehe Abbildung 55). Sollte diese IP-Adresse nicht mit der letzten IP übereinstimmen, kann diese direkt neu eingegeben werden, wie in

Abbildung 60 dargestellt, andernfalls wird die letzte IP-Adresse durch drücken von Enter übernommen. Die Kommandobefehle, die im Modus 1 verwendet werden, können ebenfalls verwendet werden. Es kommen jedoch noch ein Paar Befehle hinzu, die den Server steuern, wie z.B. „CHANGESERVER“ um den Modus des Servers wieder zu ändern.

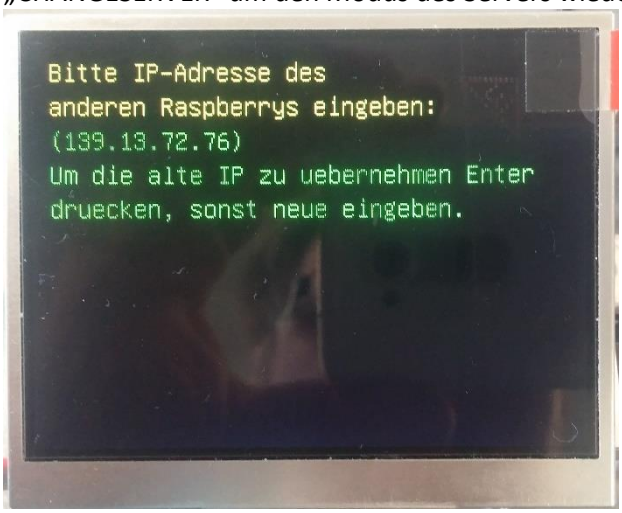

Abbildung 59 IP-Wahl

Abbildung 60 IP Neueingabe

Befehl	Funktion	Modus		
		1	2	3
EXIT	Programm wird beendet (Nur zu Programmieranwendungen!)	x		
SHUTDOWN	Eigener Raspberry wird heruntergefahren	x		x
CHANGE	Moduswechsel	x		x
CLEAR	Display leeren (Senden/Empfangen)	x		x
CONNECT	Wenn Stromschleife angeschlossen Relais AN	x	x	x
DISCONNECT	Wenn Stromschleife angeschlossen Relais AUS	x	x	x
CHANGESERVER	Moduswechsel beim Server (und beim eigenen Raspberry)		x	x
SHUTDOWNSERVER	Server herunterfahren		x	x
"	Zeichen für den Buchstabenmodus übertragen	x	x	x
$	Zeichen für den Zahlenmodus übertragen	x	x	x
#	Zeichen für die Klingel übertragen	x	x	x

Tabelle 3 Kommandobefehle Übersicht

A.6 Programm Beenden

Es gibt mehrere Möglichkeiten, das Programm zu beenden. Der Befehl „EXIT“ ist nur dann sinnvoll, wenn ein Monitor am HDMI-Port angeschlossen ist, da dieser Befehl nur das Programm beendet und nicht den Raspberry herunterfährt. Der Befehl „DISCONNECT“ wird automatisch mit ausgeführt.
Der Befehl „SHUTDOWN“ und der Menüpunkt 4 „Programm Beenden“ sind identisch:

- Das Display wird ausgeschaltet
- Der Fernschreiber wird ausgeschaltet
- Der Raspberry wird heruntergefahren

„SHUTDOWNSERVER“ führt auf dem Server den Befehl „SHUTDOWN“ aus und beim Client den Befehl „CHANGE“.

Der Fernschreiber sollte nicht zu lange (etwa eine halbe Stunde) betrieben werden, da der Vorwiderstand sehr heiß wird und anfängt zu riechen.
Leider gibt es keine Möglichkeit, dies komplett zu verhindern. Überlegungen, den Widerstand durch einen kapazitiven Blindwiderstand zu ersetzen, waren nicht erfolgreich (siehe Abschnitt 4.1 am Ende). Durch die Befehle „CONNECT“ und „DISCONNECT“ ist es allerdings möglich den Fernschreiber immer nur für kurze Zeitintervalle zu betreiben, wodurch die Wärmeentwicklung nicht so stark ausfällt.

B Anleitung Vorführplatine

B.1 Aufbau

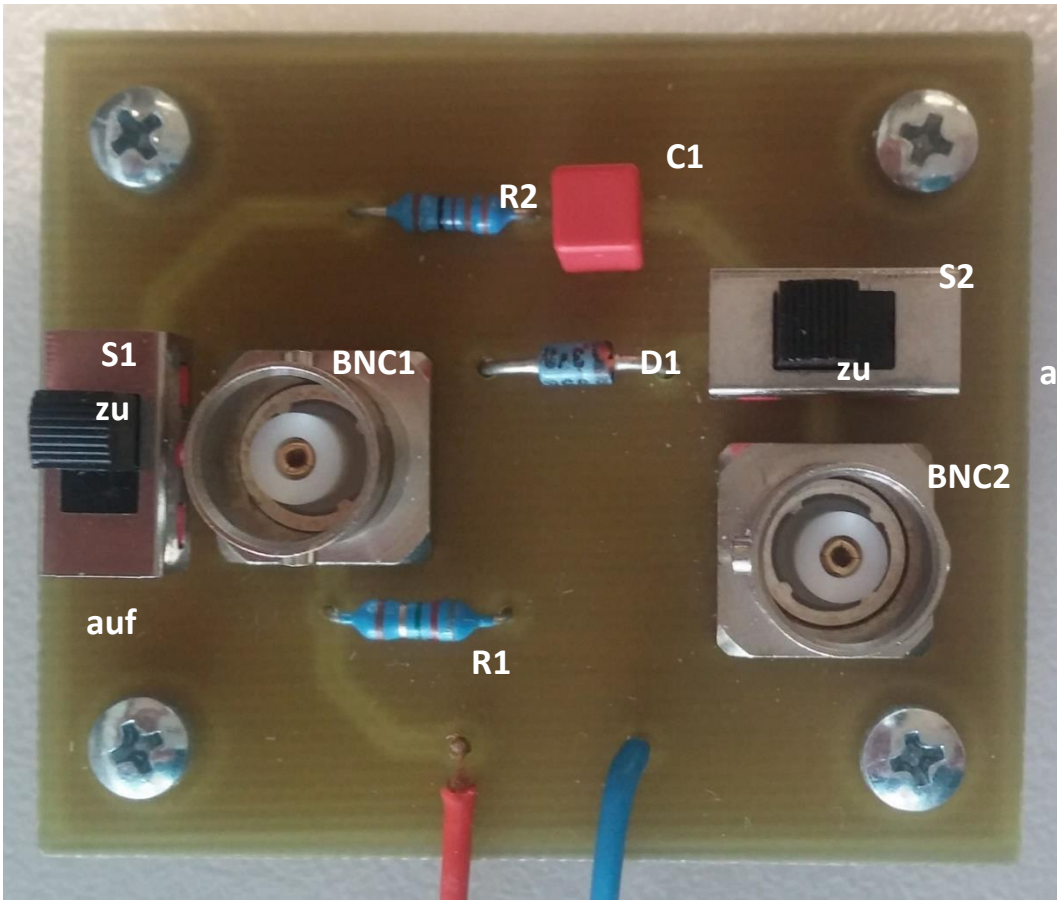

Abbildung 61 Vorführplatine

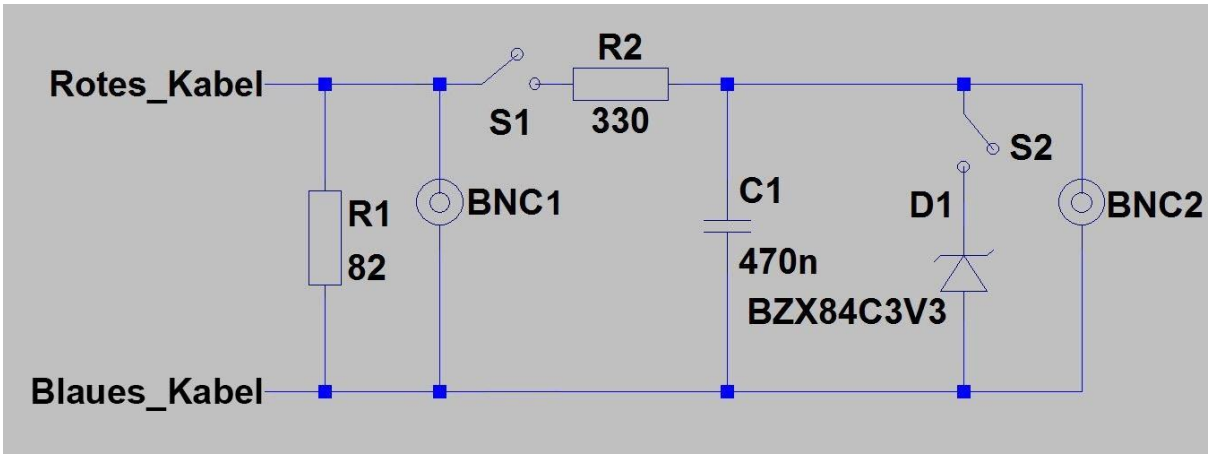

Abbildung 62 Schaltplan Vorführplatine

Die Anschlüsse an der Vorführplatine bestehen aus der roten Leitung, welche in die schwarze Buchse des Fernschreibers gesteckt wird, der blauen Leitung, welches in die schwarze Buchse des Servers gesteckt wird und 2 BNC-Buchsen, die an ein Oszilloskop angeschlossen werden können.
Man kann die Vorführplatine auch ohne Server benutzen, dazu verwendet man anstelle des Servers eine einstellbare Laborspannungsquelle und stellt den Strom auf etwa 40 mA ein, dann kann man das vom Fernschreiber kommende Signal erkennen. Wenn man den Server benutzt, kann man auch das vom Raspberry kommende Signal betrachten.

B.2 1. Teilstufe

Die erste Stufe besteht aus dem Widerstand R1, der einen Widerstandswert von 82 Ω besitzt, wodurch sich ein Strom von 38 mA einstellt, daraus folgt eine Spannung von 3,12 V über dem Widerstand. An der Buchse BNC1 lässt sich bei geöffnetem Schalter S1 das Ausgangssignal des Fernschreibers ungefiltert messen (Abbildung 63). Die Schalterstellungen sind in Abbildung 61 dargestellt. Dieses Signal ist allerdings nicht das Signal, das bei der ersten Messung, vor der Reinigung des Fernschreibers gemessen wurde, dies ist in Abbildung 64 zusehen.

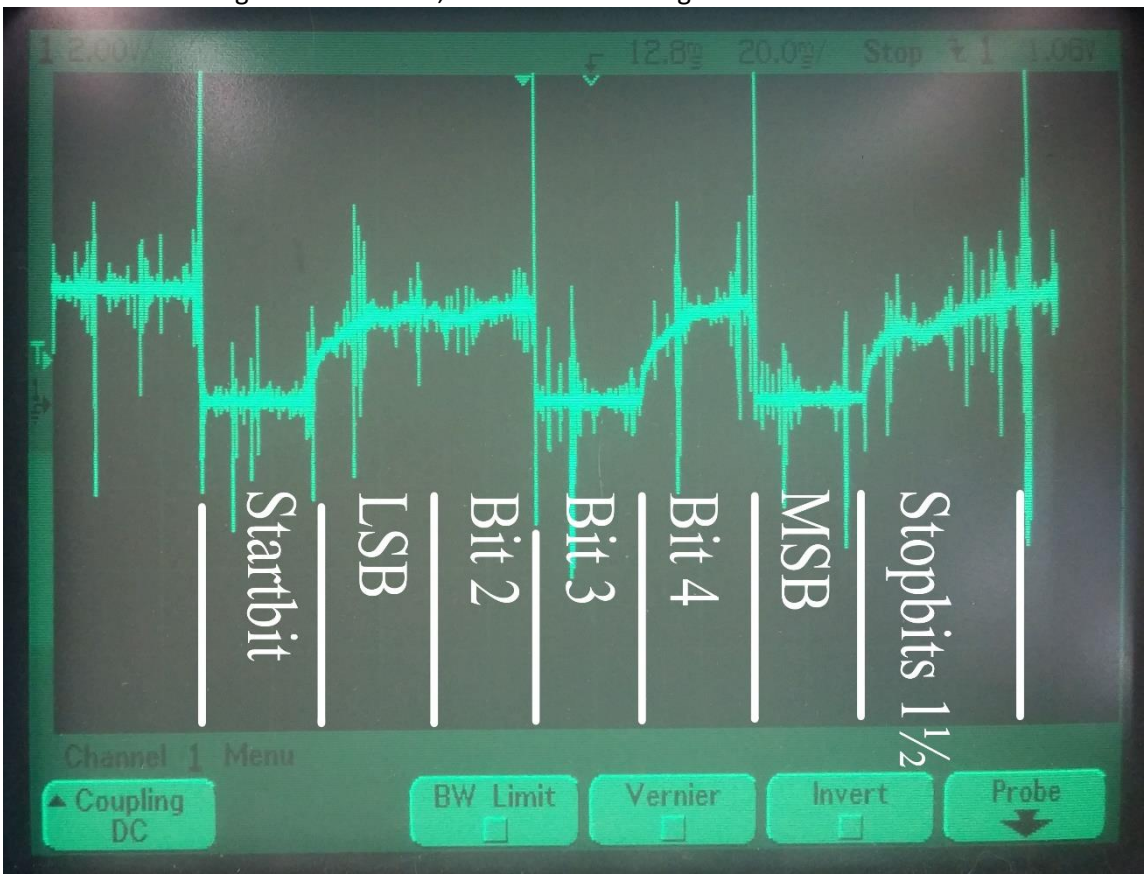

Abbildung 63 Ungefiltertes Signal, ein S wurde übertragen

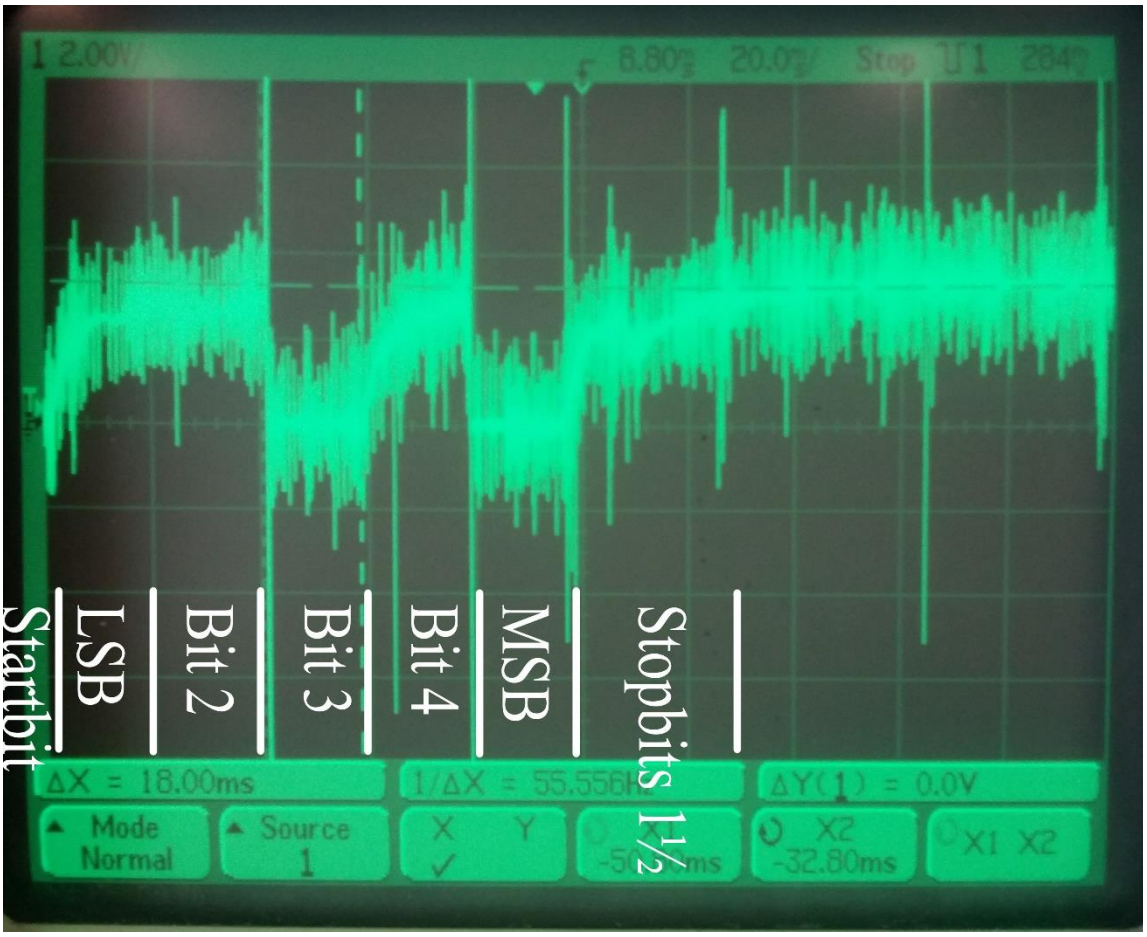

Abbildung 64 Erste Messung über einen 100 Ω Widerstand, ein S wurde übertragen

B.3 2. Teilstufe

Für den Gebrauch der 2. Teilstufe muss der Schalter S1 geschlossen und der Schalter S2 offen sein. Die zweite Stufe besteht aus dem Widerstand R2 mit 330 Ω und dem Kondensator C1 mit 470 nF, durch die Verschaltung der Bauteile ergibt sich ein Tiefpass mit einer Grenzfrequenz von 1026 Hz.

$$f_g = \frac{1}{2 * \pi * R * C} = \frac{1}{2 * \pi * 330\ \Omega * 470\ nF} = 1026\ Hz$$

Durch den Einsatz dieses Filters wird das eigentliche Signal störungsfreier, wie in Abbildung 65 dargestellt, dies ist an der Buchse BNC2 zu messen.

Abbildung 65 Signal nach Tiefpassfilter, ein S wurde übertragen

B.4 3. Teilstufe

Um die 3. Teilstufe zu nutzen, müssen die Schalter S1 und S2 geschlossen sein (siehe Abbildung 62 Schaltplan Vorführplatine“). Das Signal ist nach dem Filter zwar schon eindeutig zu erkennen, allerdings enthält es noch Spannungsspitzen, die der serielle RX-Anschluss des Raspberry Pi nicht verträgt. Daher ist eine Zenerdiode hinzugeschaltet, die die Spannung auf 3,3V begrenzt, dies ist in Abbildung 66 zu erkennen und an der Buchse BNC2 zu messen.
Bei Betrieb der 3. Teilstufe entspricht das auf dem Oszilloskop sichtbare Signal in etwa dem am RX-Pin des Raspberry Pi ankommende Signal. Man kann nicht sagen, dass die beiden Signale identisch sind, da die verwendeten Bauteile Toleranzen haben (vor allem die Zenerdiode und der Kondensator) und sich verschiedene Platinen EMV-mäßig verschieden verhalten, vor allem, da die eine in SMD-Bauweise aufgebaut wurde und die andere nicht. Außerdem ist die Platine am Raspberry Pi in einem Gehäuse.

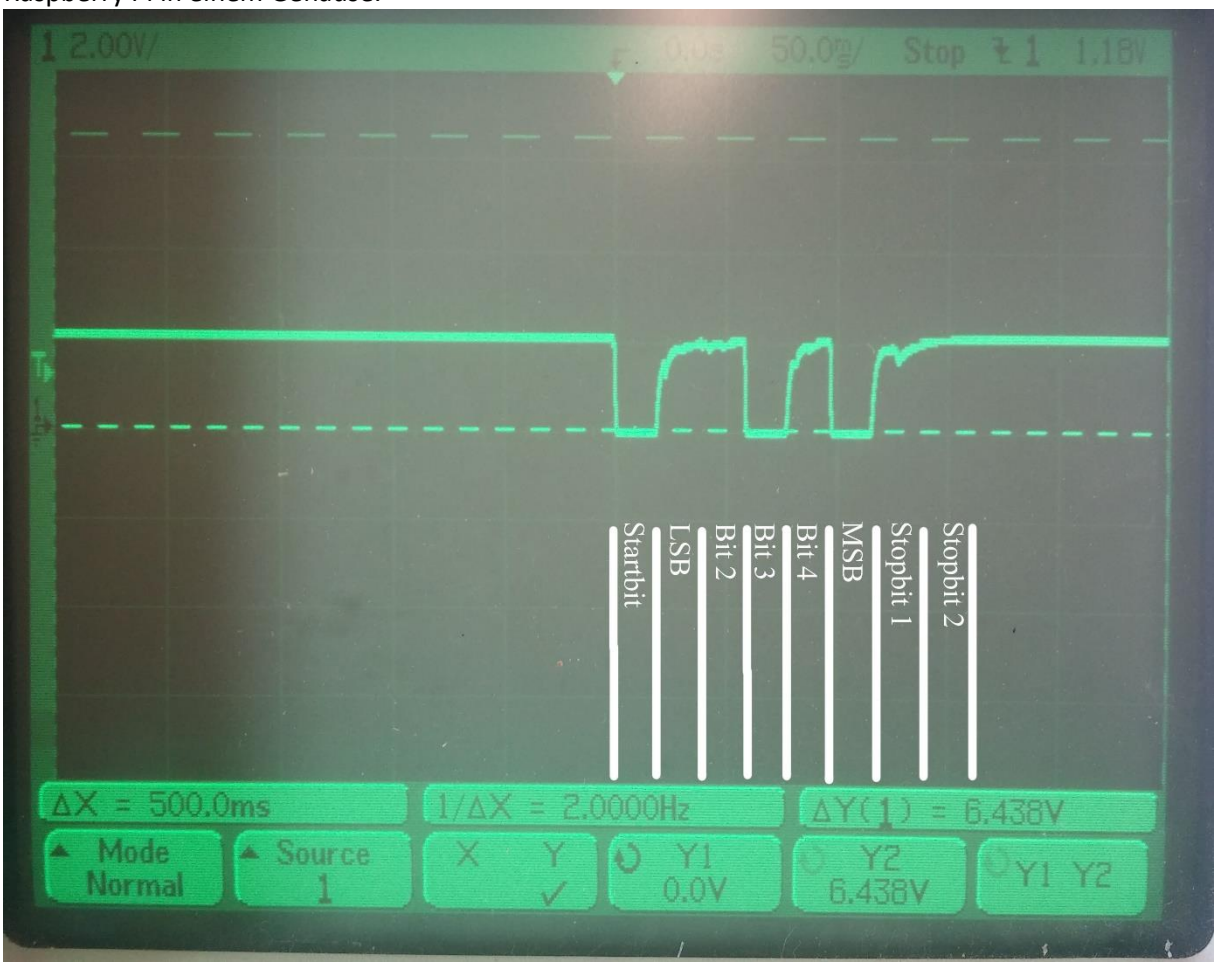

Abbildung 66 Signal an BNC2 mit Z-Diode, ein S wurde übertragen

C Anleitung zur Neuinstallation der Micro-SD-Karten

C.1 Mit Recovery-Images

C.1.1 Windows PC mit (Micro-)SD-Kartenleser

- Leere (zu überschreibende) SD-Karte einlegen, sowie die DVD „Fernschreiber Projekt Recovery“
- Server.zip oder Client.zip auf dem PC entpacken, dazu sind 16 GB freier Speicherplatz nötig
- Win32DiskImager-0.9.5-install.exe installieren und Starten
    - Auf Order-Symbol klicken
    - (DVD):\Server.img bzw. (DVD):\Client.img auswählen, darauf achten, dass als Device die SD-Karte ausgewählt ist
    - Write klicken

C.1.2 Raspberry-Pi

- SD-Karte einsetzen
- Starten

Fertig!

C.2 Ohne Recovery-Images

C.2.1 Windows PC mit (Micro-)SD-Kartenleser

- Leere (zu überschreibende) SD-Karte einlegen, sowie die DVD „Fernschreiber Projekt Neuinstallation“.
- Win32DiskImager-0.9.5-install.exe installieren und Starten
    - Auf Order-Symbol klicken
    - (DVD):\2016-05-27-raspbian-jessie.img auswählen, darauf achten, dass als Device die SD-Karte ausgewählt ist
    - Write klicken
- (DVD):\install auf einen USB-STICK kopieren

  WICHTIG!! NAME DES STICKS MERKEN, HIER ALS usb1 BEZEICHNET

C.2.2 Raspberry-Pi

- Monitor, Tastatur, Maus und USB-Stick anschließen
- Micro-SD-Karte einstecken
- Netzteil anschließen

  Der Raspberry startet nun zum ersten Mal.

- Terminal öffnen und folgende Befehle eingeben.
  - make -C /media/pi/usb1/install/
    // Taste „–“ ist auf der englischen Tastatur „/“ und „ß“ ist „-“
    //Es kommt vor, dass das Makefile „… in the future“ anzeigt, ignorieren
  - ./installieren
    - In dem gestarteten Programm werden Sie aufgerufen, dem GENAUEN Namen des USB-Sticks einzugeben (Groß- und Kleinschreibung beachten)
    - Das Programm öffnet die raspi-config, folgende Einstellungen sind vorzunehmen
      - 1 Expand Filesystem - ok
      - 5 International options
        - I1 Change Locale - de_DE.UTF-8 hinzufügen und bestätigen - de_DE.UTF-8 auswählen und bestätigen
        - I2 Change Timezone - Europe auswählen und bestätigen - Berlin auswählen und bestätigen
        - I3 Change Keyboard layout - Generic 102-key(Intl) PC - other - German - German - The default for the keyboard layout - no compose key – no
      - 9 Advanced options
      - A7 Serial – No
      - Finish und NICHT reboot
    - Es öffnet sich die Datei wpa_supplicant.conf, hier muss eine eduroam-Kennung eingegeben werden, um beim Neustart Internet zu haben. Die Daten werden im Laufe der Installation automatisch gelöscht, bleiben also nicht in der Datei stehen.
      - identity ändern
      - password ändern
      - Im Leafpad-Menü Speichern und Beenden
- reboot
- Der Raspberry führt nun einen Neustart aus
- Das Programm führt nun Updates und Installationen durch, das dauert je nach Internetverbindung recht lange.
- Danach werden Sie gebeten eine Auswahl zu treffen:
  - 1 Server Karte
  - 2 Client Karte
    Die Installation des Servers dauert etwas länger als die Client Installation.
- Der Raspberry führt nun einen Neustart aus
  Fertig.

D Weitere Bilder

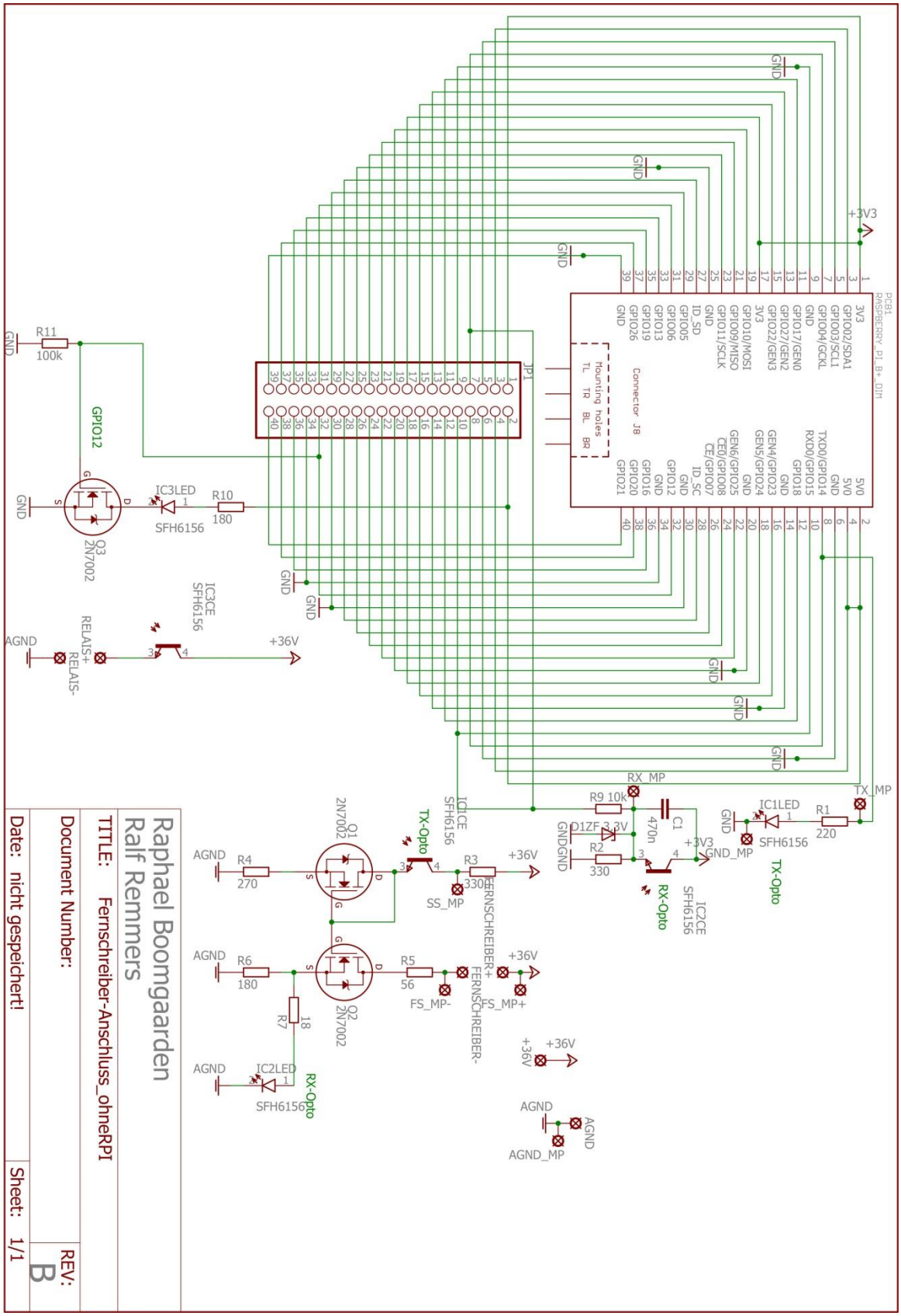

Abbildung 67 Schaltung umgesetzt in Eagle

E Abbildungsverzeichnis

Abbildung 1 Projektaufbau 2
Abbildung 2 Der verwendete Fernschreiber 4
Abbildung 3 Die einzige gefundene Bezeichnung am eigentlichen Fernschreiber 5
Abbildung 4 Typenschild des Motors 6
Abbildung 5 Verdrahtung des Fernschreibers 7
Abbildung 6 Stromlaufplan des Motors 8
Abbildung 7 Stromlaufplan des Signalpfades 9
Abbildung 8 Kondensatoren von oben 10
Abbildung 9 Kondensatoren von der Seite 11
Abbildung 10 Signal vor der Reinigung, ein S wurde übertragen 12
Abbildung 11 Signal nach der Reinigung, ein S wurde übertragen 13
Abbildung 12 Lötleiste mit Bananenbuchsen 13
Abbildung 13 Neue Platine für die Anschlussbuchsen 14
Abbildung 14: Beispiel einer Übertragung an den Fernschreiber, schematisch dargestellt 15
Abbildung 15 Fernschreiber von der Seite, Mechanik die die Codierung zeigt rot markiert 18
Abbildung 16 Der Raspberry Pi 3 Model B [4] 19
Abbildung 17 Displaydarstellung Modus 1 21
Abbildung 18 Modus 1 in Gebrauch 22
Abbildung 19 Vom Fernschreiber beschrifteter Papierstreifen 22
Abbildung 20 Displaydarstellung Modus 2 24
Abbildung 21 Modus 3 in Gebrauch 25
Abbildung 22 Inhalt des Webinterfaces mit Bild 33
Abbildung 23 Inhalt des Webinterfaces ohne Bild 33
Abbildung 24 Adapterplatine für das Display 34
Abbildung 25 Display mit Flachbandkabel 34
Abbildung 26 Beispiel für die Schematische Darstellung einer Übertragung 35
Abbildung 27 Der verwendete Thermodrucker 35
Abbildung 28 Thermodrucker von unten 36
Abbildung 29 Das Innere des Thermodruckers 36
Abbildung 30 Ausdruck des Thermodruckers 37
Abbildung 31 Schaltung der eigenen Platine für das Signal 38
Abbildung 32 Schaltung zur Ansteuerung des Relais 39
Abbildung 33 Schaltung in Eagle 40
Abbildung 34 Layout mit Vorlage 41
Abbildung 35 Layout ohne Vorlage 42
Abbildung 36 Schaltplan für die Relaisplatine 43
Abbildung 37 Platinenlayout der Relaisplatine 44
Abbildung 38 Ansicht des Gehäuses ohne Deckel 46
Abbildung 39 Gehäuse von hinten 46
Abbildung 40 Gehäuse von der rechten Seite 47
Abbildung 41 Gehäuse von oben 47
Abbildung 42 Gehäuse von vorne 48
Abbildung 43 Erste Messung des Fernschreibersignals, ein S wurde übertragen 49
Abbildung 44 Das gefilterte Signal, ein S wurde übertragen 50

Abbildung 45 Vom Raspberry Pi ausgehendes Signal, ein S wurde übertragen 51
Abbildung 46 Signal im RX-Pfad mit offenem Deckel, ein S wurde übertragen 51
Abbildung 47 Signal im RX-Pfad mit geschlossenem Deckel, ein S wurde übertragen 52
Abbildung 48 Rückseite Servergehäuse 55
Abbildung 49 Rechte Seite Servergehäuse 55
Abbildung 50 Anschlussplatine Fernschreiber 55
Abbildung 51 Auswahlmenü Server 56
Abbildung 52 Auswahlmenü Client 56
Abbildung 53 Drucker Abfrage 57
Abbildung 54 Modus 1 57
Abbildung 55 Modus 2 58
Abbildung 56 Anzeige.php 58
Abbildung 57 Anzeige2.php 59
Abbildung 58 Modus 3 59
Abbildung 59 IP-Wahl 60
Abbildung 60 IP Neueingabe 60
Abbildung 61 Vorführplatine 62
Abbildung 62 Schaltplan Vorführplatine 62
Abbildung 63 Ungefiltertes Signal, ein S wurde übertragen 63
Abbildung 64 Erste Messung über einen 100 Ω Widerstand, ein S wurde übertragen 64
Abbildung 65 Signal nach Tiefpassfilter, ein S wurde übertragen 65
Abbildung 66 Signal an BNC2 mit Z-Diode, ein S wurde übertragen 66
Abbildung 67 Schaltung umgesetzt in Eagle 69

F Tabellenverzeichnis

Tabelle 1 Vergleich Baudot-Code und Code des vorliegenden Fernschreibers 17
Tabelle 2 Übersicht über die Befehlscodes, die in den verschiedenen Modi verfügbar sind 26
Tabelle 3 Kommandobefehle Übersicht 60

G Codeabschnittverzeichnis

Codeabschnitt 1 Initialisierung der seriellen Schnittstelle 28
Codeabschnitt 2 Übersetzung von ASCII-Zeichen in Baudot-Code (Auszug) 29
Codeabschnitt 3 Übersetzung von Baudot-Code in ASCII-Zeichen (Auszug) 31

H Literaturverzeichnis

[1] F. Wittendorfer, „Siemens-Fernschreiber T 24,“ Siemens, Berlin, 2017.

[2] H. W. Pendry, in *The Baudôt Printing Telegraph System. Second Edition*, London, Sir Isaac Pitman & Sons, Ltd., 1919, pp. 43-44.

[3] E. Upton, „Raspberry Pi 3 on sale now at $35 - Raspberry Pi,“ Raspberry Pi Foundation, 29 Februar 2016. [Online]. Available: https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/. [Zugriff am 6 09 2016].

[4] S. Petersen, „File:9815 - Raspberry Pi 3.jpg - Wikimedia Commons,“ 5 März 2016. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=47335669. [Zugriff am 5 September 2016].

[5] J. Adams, „rpi-3b-reduced - RPI-3B-V1_2-SCHEMATIC-REDUCED.pdf,“ 4 April 2016. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/RPI-3B-V1_2-SCHEMATIC-REDUCED.pdf. [Zugriff am 6 September 2016].

[6] I. T. Storm, „Raspberry Pi 3 mit ARMv8 ist da | heise online,“ Heise Online, 29 Februar 2016. [Online]. Available: http://www.heise.de/newsticker/meldung/Raspberry-Pi-3-mit-ARMv8-ist-da-3119692.html. [Zugriff am 6 September 2016].

[7] Raspberry Pi Foundation, „Raspberry Pi 3 Model B - Raspberry Pi,“ [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Zugriff am 6 September 2016].

[8] Conrad Electronic SE , „Raspberry Pi® Display-Modul C-Berry Raspberry Pi® auf conrad.de online bestellen | 001020910,“ [Online]. Available: https://www.conrad.de/de/raspberry-pi-display-modul-c-berry-raspberry-pi-1020910.html. [Zugriff am 26 September 2016].

[9] Adafruit Industries, „Mini Thermal Receipt Printer ID: 597 - $49.95 : Adafruit Industries, Unique & fun DIY electronics and kits,“ [Online]. Available: https://www.adafruit.com/products/597. [Zugriff am 27 09 2016].

[10] T. R., „Eagle LBR: Raspberry Pi B+ - Mikrocontroller.net,“ 18 Juli 2014. [Online]. Available: http://www.mikrocontroller.net/topic/339008. [Zugriff am 19 September 2016].

Email vom 03.02.2017 von Dr. Frank Wittendorfer an Raphael Boomgaarden:

Betreff: Siemens-Fernschreiber T 24

„Hallo Herr Boomgaarden,

gar nicht so einfach. Es handelt sich um das Modell „T 24“. Danach wird man suchen müssen. Mit „alpha. 29“ ist die Tastatur gemeint. Dies erschließt sich aus unserer produktgeschichtlichen Sammlung, in der sich ein Fernschreiber mit expliciter Tastatur „alpha. 34“ befindet.

Ein T 24 ist auch hier abgebildet:
http://media.gettyimages.com/photos/fernschreiber-modell-der-firma-siemens-halske-1931-picture-id542416171

Dies gibt einen weiteren Hinweis. Das Gerät scheint, wenn die Information stimmt, aus dem Jahr 1931 zu stammen und ist somit ein recht frühes Modell.

Leider ist genau über diesen T 24 nicht nur in unserem Archiv, sondern auch im Internet nichts zu finden. Er ist offensichtlich auch in keiner der kleinen privaten Fernschreibersammlungen vertreten.

Die Siemens-Fernschreiber wurden von der „Siemens & Halske AG“ vertrieben, der nachrichtentechnischen Sparte unseres Hauses. In den Geräten verbaut wurden Motoren der „Siemens-Schuckertwerke AG“, der starkstromtechnischen Sparte.

Dies ist alles, was sich im Moment dazu sagen läßt. Ich werde mich nächste Woche gerne noch mit einer Expertin aus unserem Team dazu besprechen. Da ist noch ein wenig Hoffnung …

Freundliche Grüße aus Berlin
Frank Wittendorfer

Mit freundlichen Grüßen
Dr. Frank Wittendorfer
Leiter Siemens-Archiv
Siemens AG
Communications
Siemens Historical Institute
CC SHI
Nonnendammallee 101 / Besuchereingang Rohrdamm 83
13629 Berlin, Deutschland
Tel.: +49 30 386-55881
Mobil: +49 173 5249084
mailto:frank.wittendorfer@siemens.com
www.siemens.com/ingenuityforlife

Siemens Aktiengesellschaft: Vorsitzender des Aufsichtsrats: Gerhard Cromme; Vorstand: Joe Kaeser, Vorsitzender; Roland Busch, Lisa Davis, Klaus Helmrich, Janina Kugel, Siegfried Russwurm, Ralf P. Thomas; Sitz der Gesellschaft: Berlin und München, Deutschland; Registergericht: Berlin Charlottenburg, HRB 12300, München, HRB 6684; WEEE-Reg.-Nr. DE 23691322“ [1]