



Bachelorthesis

im Wintersemester 2015/2016

Fachbereich 2 Informatik und Ingenieurwissenschaften

der Frankfurt University of Applied Sciences

im Studiengang

Informatik Bachelor of Arts (B.A.)

vorgelegt von

Marko Mihaljevic

geb. am 14.03.1988 in Frankfurt am Main

Matrikelnummer: 878834

Thema:

Ansteuerung einer in Reihe geschalteten LED-Matrix mit Hilfe eines Einplatinencomputers

Referent:

Prof. Dr. Christian Baun

Korreferent:

Prof. Dr. Thomas Gabel

Arbeit abgegeben am: 29.03.2016

Eidesstattliche Erklärung

Hiermit erkläre ich,

Mihaljevic, Marko, geboren am 14.03.1988 in Frankfurt am Main,

dass ich die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Soweit ich auf fremde Materialien, Texte oder Gedankengänge zurückgegriffen habe, enthalten meine Ausführungen vollständige und eindeutige Verweise auf die Urheber und Quellen. Alle weiteren Inhalte der vorgelegten Arbeit stammen von mir im urheberrechtlichen Sinn soweit keine Verweise und Zitate erfolgen.

Mir ist bekannt, dass ein Täuschungsversuch vorliegt, wenn die vorstehende Erklärung sich als unrichtig erweist.

Die aufgeführte Arbeit ist in gleicher oder ähnlicher Fassung noch nicht als Bestandteil einer Studien- oder Prüfungsleistung erschienen.

Ort, Datum

Unterschrift

Danksagung

Zu jeder Abschlussarbeit gehört auch eine Danksagung. Danken möchte ich mich erst einmal bei allen, die mich während der Studienzeit sowie der Anfertigung dieser Abschlussarbeit stets tatkräftig unterstützt und motiviert haben.

Mein größter Dank geht an meine Eltern, die mir auch in schweren Situationen beistanden und mir das Studium und bisherigen Werdegang ermöglicht haben.

Ganz besonders danke ich Prof. Dr. Christian Baun für die Bereitstellung des Themas und die Unterstützung während der Anfertigung dieser Abschlussarbeit. Ebenso geht ein Dank an Prof. Dr. Thomas Gabel, der sich bereit erklärt hat als Zweitprüfer diese Arbeit zu betreuen.

Danke!

Marko Mihaljevic

Zusammenfassung

Die vorliegende Bachelorthesis beschreibt die Ansteuerung sowie die Realisierung einer in Reihe geschalteten LED-Matrix mit jeweils 16x32 LEDs durch den Einplatinencomputer Raspberry Pi. Hierzu werden die GPIO-Anschlüsse des Raspberry Pi mit der LED-Matrix gekoppelt und entsprechend grafische, sowie textbasierte Ergebnisse über eine Ansteuerungsbibliothek erzielt. Die Funktionsweise zwischen den einzelnen Komponenten soll im weiteren Verlauf geklärt und verdeutlicht werden. Das Resultat dieser Arbeit ergibt sich aus der erfolgreichen Ansteuerung und Darstellung verschiedener Elemente auf der LED-Matrix.

Abstract

The following thesis deals with a lined up LED-matrix with each 16x32 LEDs, controlled and executed by the single-board computer Raspberry Pi. For this purpose the board's GPIO-connectors will be linked to the LED-matrix. In order to achieve graphical, and as well textual output, certain control libraries will be applied. Thus the functionality between the components will be clarified. The successful display of different elements via the led-matrix will be the ultimate result of this paper.

Inhaltsverzeichnis

1. Einführung	8
1.1. Hintergrund und Motivation	8
1.2. Zielsetzung	9
1.3. Themeneingrenzung und Aufbau.....	10
2. Technische Grundlage	11
2.1. Das Grundprinzip des Raspberry Pi.....	11
2.2. Raspberry Pi Modell B+	12
2.3. GPIO Schnittstelle	14
2.4. LED-Matrix	15
2.5. Verfügbare Distributionsvarianten	17
2.6. Vergleichseinschätzung zu anderen Einplatinencomputer	19
3. Vorbereitung und Installation.....	22
3.1. Hardwarekomponenten	22
3.2. Aufbau und Verbindung der LED-Matrix mit dem Raspberry Pi.....	25
3.3. Imagevorbereitung.....	29
4. Konfigurationsschritte	32
4.1. Starten des Bootvorgangs	32
4.1.1. Vorgehensweise bei der raspi-config	32
4.1.2. Funktionsweise Konsolenbefehl <i>sudo</i>	37
4.1.3. Fernzugriff über SSH.....	38
4.2. Update	41
4.3. Raspberry Pi-update.....	41
4.4. Paketverwaltungssystem apt	43
5. Einrichtung Software und Programmcode	45
5.1. Export Bash	45
5.2. Export via wiringpi2	47
6. GPIO-Ansteuerung	49
6.1. Beispielschreibprozess unter C/C++	49
7. Informationsausgabe mittels LED-Matrix.....	52
7.1. Testlauf über LED-Matrix mit C.....	52
7.2. Testlauf über LED-Matrix mit Python.....	55

9. Resümee und Ausblick	70
Literaturverzeichnis.....	72
Abbildungsverzeichnis.....	82
Abkürzungsverzeichnis	84

1. Einführung

1.1. Hintergrund und Motivation

Das enorme Voranschreiten der technologischen Entwicklung¹, erlaubt eine fortschreitende Verbreitung von Einplatinencomputer auf dem Markt. Indessen hat sich ein Anwender als besonders populär² erwiesen, der sogenannte Raspberry Pi³. Die Erfindung des Briten Eben Upton, sollte zunächst als Unterrichtselement für britische Schüler dienen, jedoch stellt sich bald ein ausschlaggebender Erfolg⁴ zu Tage. Besonders seit 2012 hat der Linux Einplatinencomputer nun eine starke Präsenz in der IT-Welt eingenommen und ist neben dem äußerst zuvorkommenden Preis auch mit seinen vielfältigen Einsatzmöglichkeiten zu einem wichtigen Element in der Branche geworden. So kann das kreditkartengroße Gerät unter anderem als Spielekonsole⁵, NAS, Radio, Medien-Center oder als Mini-Rechner⁶ eingesetzt werden. Brauchte man in der Vergangenheit noch ganze Räume mit Großrechnern⁷, um diverse Rechenoperationen durchzuführen, ist das heutzutage dank moderner Prozessor Chips zu einer trivialen Gegebenheit in unzähligen Geräten⁸ geworden. Der Computer heute muss ganz neue und unterschiedliche Anforderungen erfüllen. Infolgedessen werden immer mehr Funktionalitäten auf dem Computer-Prozessor verbaut, wobei die Größe des Gerätes abnimmt. Unser Alltag ist geprägt von Computern und sowohl in den großen Branchen, als auch im Privatgebrauch nicht mehr wegzudenken⁹. In Verbindung mit anderen elektronischen Bauteilen können so unverwechselbare Systeme entstehen, die den Alltag der Menschen in vielerlei Hinsicht erleichtern. So ist auch der Gebrauch von LED-Matrizen in vielen Lebensbereichen gängig. Sie finden sich im Verkehr¹⁰ oder auch als Informationsquellen für Allerlei wieder¹¹. Der dafür vorgesehene Input bzw. die Ansteuerung solcher LED-Matrizen, lässt sich eben auch durch den kleinen Alleskönner Raspberry Pi realisieren. Insbesondere die GPIO

¹ Vgl. Gartner, Inc.: Hype Cycle 2015

² Vgl. Marshall Gary

³ Vgl. Raspberry Pi Foundation

⁴ Vgl. Kling, Bernd

⁵ Vgl. Dragicevic, Marko

⁶ Vgl. Rixecker, Kim

⁷ Vgl. Michael, Georg

⁸ Vgl. Hackbarth, Andrea

⁹ Vgl. Mattern, Friedemann

¹⁰ Vgl. B.A.S. Verkehrstechnik AG

¹¹ Vgl. Computronics Ltd.

Schnittstelle verhilft dem Gerät zu seinem facettenreichen Auftreten. Da in der heutigen Zeit besonders der Kosten- und Energiesparaspekt¹² wichtige Faktoren in der IT-Welt sind, ist der Raspberry Pi gefragter denn je. Der Drang sich selbst mit dieser Thematik auseinanderzusetzen ist aus der Faszination entstanden, einen kleinen und doch vollwertigen Rechner zu besitzen, welcher sich für jegliche Art von Projekt eignet und dabei ein äußerst niedriges Preisniveau¹³ besitzt. Begeisterung ist auch durch die immense Unterstützung und dem Austausch der Online Community entstanden¹⁴. Die Auswahl den Raspberry Pi mit einer LED-Matrix zu verknüpfen hat sich in erster Linie aus dem Erstaunen gebildet, solch ein Projekt für private Zwecke, so beispielsweise für Werbereklamen¹⁵, einzusetzen. Explizit haben die vorteilhaften Eigenschaften des Raspberry Pi zur Verwendung in dieser Arbeit geführt. Neben den Anschaffungskosten, sprechen auch die Folgekosten des kleinen Einplatinencomputers, sowie der geringe Stromverbrauch¹⁶ und der Geräuschpegel für den Einsatz.

1.2. Zielsetzung

Ziel der vorliegenden Arbeit soll es unter anderem sein, eine erfolgreiche Ansteuerung einer LED-Matrix durch den Einplatinencomputer Raspberry Pi aufzuzeigen. Gleichzeitig soll eine günstige Open-Source-Lösung herausgearbeitet werden, die Werbe bzw. Informationstafeln für den Normalverbraucher¹⁷ ermöglicht. Solch ein Projekte soll auch aufzeigen, welche kostengünstigen Möglichkeiten sich Studenten im Rahmen ihrer Studienzeit bieten, um so beispielsweise Programmierkenntnisse zu vertiefen. Aber auch die Untersuchung theoretischer Ansätze im öffentlichen Bereich soll kurz miteinbezogen werden.

¹² Vgl. Itdz-berlin

¹³ Vgl. raspishop.de

¹⁴ Vgl. Raspberry Pi Foundation

¹⁵ Vgl. TABLICE LED, TELEBIMY

¹⁶ Vgl. Elektronik-Kompendium.de

¹⁷ Vgl. TABLICE LED, TELEBIMY

1.3. Themeneingrenzung und Aufbau

Die Präsenz der Einplatinencomputer hat sich in den letzten Jahren stark verbreitet. Neben dem Raspberry Pi gibt es zahlreiche Alternativen und Lösungen, die an dieser Stelle nicht alle genannt, jedoch in verschiedenen Abschnitten erwähnt werden können. Der Hauptaspekt dieser Arbeit soll sich auf den Realisierungsvorgang durch den Raspberry Pi konzentrieren. Gleichzeitig wird auf eine erfolgreiche Informationsausgabe auf der LED-Matrix hingearbeitet. Nebestehend sollen alle erforderlichen Tools und Softwarekomponenten untersucht und in den Entstehungsprozess integriert werden. Entstehende Problemstellungen können über Alternativen gelöst oder bei nicht erfolgreicher Beseitigung analysiert werden und zum weiteren Bestandteil dieser Arbeit werden. Weiterhin wird in dieser Thesis ermittelt, warum sich gerade der Raspberry Pi optimal für den Realisierungsaufbau eignet, bzw. ob andere Einplatinencomputer breitgefächerte Funktionalitäten in diesem Umfeld bieten. Der Ansteuerungsprozess über den Raspberry Pi selbst soll jedoch im Zentrum der Aufmerksamkeit bleiben und wird im Laufe der Thesis näher erläutert. Zudem soll eine abschließende Schlussbetrachtung der gesamten Arbeit erfolgen. Darin werden nochmals in Kürze die wichtigsten Aspekte für den Bearbeitungsprozess aufgegriffen. Darüber hinaus lässt sich die nun vorliegende Arbeit sowohl als Leitfaden aber auch als Versuchsprotokoll des zu behandelnden Themas verstehen.

2. Technische Grundlage

Das nachstehende Kapitel wird sich zunächst auf grundlegende Aspekte rund um den Raspberry Pi konzentrieren. Dabei werden die verwendeten Hardwareelemente vorgestellt und entsprechende Begrifflichkeiten geklärt.

2.1. Das Grundprinzip des Raspberry Pi

Wie bereits angerissen, ist der Raspberry Pi der britischen Foundation ein kreditkartengroßer und preisgünstiger Einplatinencomputer, der mit der Absicht entwickelt worden ist, den Zugang zu Programmier- und Hardwarekenntnissen zu erleichtern¹⁸.

Die ersten Konzeptionen für ein Gerät dieser Art wurden 2006 auf Grundlage eines Atmel ATmega664 Micro Controller geschaffen¹⁹. Zwar kann der soeben genannte Micro Controller durch seinen einfachen Aufbau und der relativ zugänglichen Programmierbarkeit punkten, doch in Anbetracht der Leistung ist es noch kein Durchbruch. Als dann im Jahr 2010 durch den Smartphone-Boom²⁰ eine sogenannte „System on A Chip“²¹ Architektur auf den Markt kommt, entsteht mittels der verbauten Grafikeinheit, dem Arbeitsspeicher und unterschiedlichen Schnittstellen, sowie der relativ hohen Leistung, das Grundkonzept für den Raspberry Pi²².

Darauf aufbauend, entsteht das erste Modell des Raspberry Pi und wird 2011²³ in den Umlauf gebracht. Der Durchbruch allerdings gelingt erst im Jahr 2012²⁴ mit einer verbesserten Version, die unter anderem eine neue GPIO-Belegung und zwei Befestigungslöcher beherbergt. Der Erfolg²⁵ des Einplatinencomputers lässt sich aber nicht nur durch seine konventionellen Hardwareaufbau erschließen, sondern auch durch die vielfältigen Installationsmöglichkeiten von Betriebssystem und der Akzeptanz verschiedener Programmiersprachen²⁶. Im Laufe der Zeit ist durch die immense Beliebtheit des Raspberry die Nachfrage drastisch angestiegen²⁷. Das hat die

¹⁸ Vgl. Raspberry Pi Foundation

¹⁹ Vgl. Raspberry Pi Foundation

²⁰ Vgl. Arthur, Charles

²¹ Vgl. Damani, Viranch

²² Vgl. Raspberry Pi Foundation

²³ Vgl. Raspberry Pi Foundation

²⁴ Vgl. Raspberry Pi Foundation

²⁵ Vgl. Raspberry Pi Foundation

²⁶ Raspberry Pi: Das umfassende Handbuch

²⁷ Vgl. Woody Roy

Entwickler dazu bewegt neue, abgeänderte Modelle vorzustellen, die den Anforderungen gerecht werden.

2.2. Raspberry Pi Modell B+

Wie im vorhergehenden Kapitel erwähnt, sind bereits einige Modelle des Raspberry Pi auf dem Markt, das Hauptaugenmerk für diese Arbeit wird jedoch das Model B+ (ABB1) sein, da es zahlreiche Erweiterungen und neue Funktionen, gegenüber seinen Vorgängern bietet²⁸. Genauerer Begutachten des Einplatinencomputers bringt als Herzstück einen BCM2835 hervor, welcher mit 700MHz getaktet ist. Zum Vergleich²⁹ beinhaltet die CPU in einem iPhone 4S 800MHz. Das Besondere hieran ist die ARM-Architektur der CPU, da sie bei effektiver Leistungsbringung, energieschonend bleibt. Indessen unterscheidet sich diese Art der CPU-Architektur maßgeblich von einer x86 bzw. x64-Architektur in einem standardmäßigen Desktop PC³⁰. Daher ist eine Installation von Software für einen normalen Desktop PC auf dem Einplatinencomputer Raspberry Pi nicht möglich und führt auch in umgekehrter Richtung, ohne den Einsatz von Emulatoren³¹, zu keinem Erfolg.

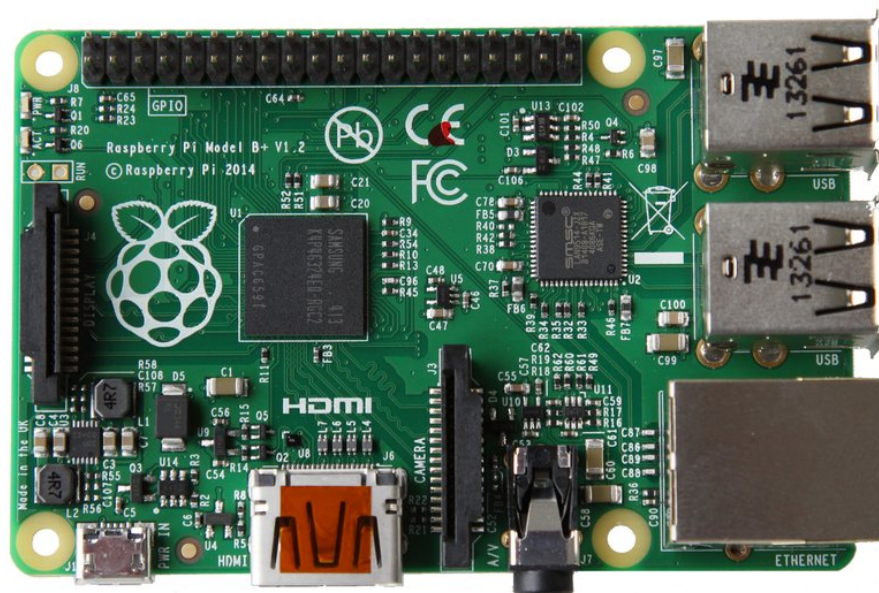


Abbildung 1 Raspberry Pi Platine

²⁸ Vgl. Kofler, Michael

²⁹ Vgl. devicespecifications.com

³⁰ Vgl. Anthony, Sebastian

³¹ Vgl. Raspberry.tips

Da der Raspberry Pi als grundlegendes Aufbaukonzept ein sogenanntes System on a Chip Prinzip (SoC) verwendet, ist die Unterbringung verschiedener Komponenten und damit auch diverser Funktionen nichts Ungewöhnliches.

Die weitere Zusammensetzung³² im Raspberry Pi Model B+ beläuft sich aus einer Dual-Core GPU mit 250 MHz Takt und integriertem 1080p H2 64 Video En/Decoder. Enthalten ist zudem 512 MB RAM, USB Schnittstellen, GPIO-Pins und eine Kameraschnittstelle. Lediglich auf einen Anschluss für Festplatten wird hier verzichtet. Stattdessen bietet sich die Möglichkeit über einen micro-SD Karten-Slot, das Betriebssystem direkt zu booten. Der Unterschied^{33, 34} zum Vorgänger ist zum einen die Anzahl der GPIO-Pins und der USB-Anschlüsse sowie einer Leistungssteigerung und einem micro-SD Karten-Slot.

Prozessor	Broadcom BCM2835
Arbeitsspeicher	512 MB
CPU	ARM1176JZF-S (700 MHz)
GPU	Broadcom Dual Core VideoCore IV
LAN	10/100 Mbit/s
USB-Anschluss	4 x USB 2.0
Videoausgabe	HDMI, Composite Video
Audioausgabe	Klinkenstecker 3,5 mm, HDMI
Schnittstellen	40 GPIO-Pins, I2C, SPI und UART, CSI, DSI, microSD Slot
Maße	85,6 mm x 56 mm x 20 mm (Länge x Breite x Höhe)
Gewicht	ca. 45 g
Stromversorgung	5V via Micro-USB Anschluss
Stromverbrauch	ca. 500 - 600 mA (2,5-3W)

Tabelle 1 Raspberry Pi Modell B+ Datenblattübersicht

³² Vgl. Adafruit Industries

³³ Vgl. Elektronik-Kompendium.de

³⁴ Vgl. Hammerschmit, Anton

2.3. GPIO Schnittstelle

Der Raspberry Pi besitzt eine spezielle Eigenschaft, welche den Einplatinencomputer von einem normalen PC unterscheidet. Das ist die sogenannte GPIO-Schnittstelle, bestehend aus Ein- und Ausgabepins, erlaubt sie das Verbinden und Ansteuern verschiedener Hardwareelemente³⁵. Durch die autarke Programmierbarkeit³⁶ und dem Senden elektrischer Signale über den ARM Mikroprozessor BCM2835, ist es möglich Daten auf allerlei Gerätschaften zu übertragen³⁷. Auf diese Weise können auch kleine Motoren, Displays, LEDs, Sensoren und Aktoren befehligt werden³⁸. Anlässlich der Diversität in der Pin-Belegung, sind somit vielseitige Funktionen möglich. Auf dem Einplatinencomputer tummeln sich neben den 26 allgemeinzweckmäßigen GPIO-Pins für die Ein- und Ausgabe, auch zwei 3,3 Volt und 5 Volt Pins für die Versorgungsspannung, sowie 8 Masse Pins (0V). Abgesehen davon existieren noch 2 DNC Pins, welche jedoch für den normalen Benutzer keine Funktionalität erfüllen³⁹. Sie dienen lediglich den Entwicklern für das Freischalten spezieller Eigenschaften⁴⁰. Somit setzt sich für das Modell B+ eine Pin-Belegung von 40 Stück zusammen. Das heißt einige der genannten Pins erfüllen eine spezielle vorkonfigurierte Funktion, doch trotzdem bleiben sie frei programmierbar. Hieraus folgend ist ersichtlich, dass unterschiedliche Arten von Projekten eine gewisse Vorüberlegung bei der Anschlussauswahl erfordern. Zudem ist beim erhöhten Einsatz von Spannungen eine Beschädigung des Einplatinencomputers nicht auszuschließen⁴¹. Diese ungemeine Flexibilität der GPIO-Pins bewilligt in Kombination mit verfügbarer Software das Integrationspotential⁴² des Einplatinencomputers in andere Produkte und Automaten. Die unten stehende Grafik (ABB2) bildet die Schnittstelle in einer vereinfachten Form dar. Bei der Zählung der Pins wird von links nach rechts und von oben nach unten geschaut. Um den ersten Pin auswendig zu machen, dreht man den Raspberry Pi einmal nach hinten. Hier ist zu erkennen, dass genau ein Pin eine eckige Lötstelle⁴³ besitzt. Die mit der Spannung beschrifteten Pins, können bis zu der angegebenen Volt-

³⁵ Vgl. Ziegelwanger, Werner

³⁶ Vgl. Dragicevic, Marko

³⁷ Vgl. Tutorials-RaspberryPi

³⁸ Vgl. ebd.

³⁹ Vgl. MATT

⁴⁰ Vgl. ebd.

⁴¹ Vgl. Elektronik-Kompendium.de.

⁴² Vgl. Christian Klaub

⁴³ Vgl. Schwarz, Andreas

Höhe mit Strommodulen versorgt werden. Pins mit der Bezeichnung *GND*⁴⁴ fungieren als leitende Körper mit der Masse 0 Volt. Wohingegen die Pins mit der Aufschrift *GPIO*⁴⁵ für bereits genannte Ein- und Ausgabe verwendet werden dürfen. Erwähnenswert sind auch noch die I²C- und SPI-Schnittstellen⁴⁶. I²C ist ein serieller Datenbus, der die Kommunikation zwischen zwei Schaltungsebenen erlaubt. Das ist insofern wichtig, da viele Embedded-Systeme, wie beispielsweise Sensoren und Aktoren über solche I²C-Buse angebunden werden müssen. Das SPI bietet, ähnlich wie der I²C Bus, eine Möglichkeit zur Anschließung von Mikrocontrollern. Unter Verwendung des Master-Slave-Prinzips, können so verschiedenen Elemente miteinander verbunden werden.

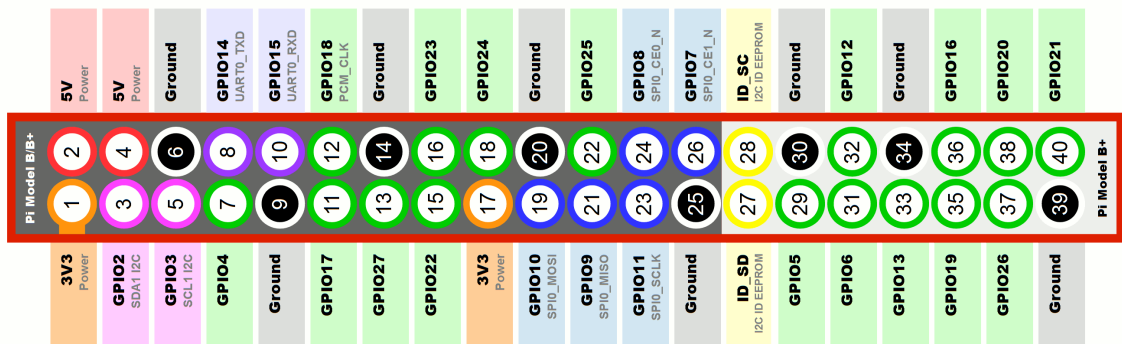


Abbildung 2 GPIO Pin-Belegung

2.4. LED-Matrix

Die LED-Matrix ist ein wichtiger Bestandteil dieser Arbeit. Um ihre Funktionsweise und Brauchbarkeit für die Realisierung nachzuvollziehen, wird dieses Element näher beleuchtet. Die Matrix insgesamt ist als eine Art Gitter aus einzelnen LEDs vorzustellen. Das Besondere hieran ist, dass jede LED-Lampe einzeln ansprechbar ist und im Grunde genommen mit wenig Aufwand viele Lampen ansprechbar sind. Damit die LEDs überhaupt leuchten können, sind sie jeweils über Kathoden (-) und Anoden (+) verbunden⁴⁷. Die Anordnung erfolgt in Spalten und Zeilen. Jedoch ist die Belegung nicht maßgebend. Die im weiteren Verlauf aufkommende Ansteuerung der LEDs

⁴⁴ Vgl. Ryniker, Richard

⁴⁵ Vgl. Ryniker, Richard

⁴⁶ Vgl. Ryniker, Richard

⁴⁷ Vgl. Schwarz, Andreas

erfolgt über das sogenannte Multiplexing⁴⁸. Es ist möglich sowohl Zeilen als auch Spalten zu multiplexen. Die Funktionsweise kann nun wie folgt dargelegt werden. Beim Anschließen der Matrix herrscht im sogenannten Grundzustand⁴⁹ keine positive Spannung für die Zeilen und GND der Spalten. Für die Beleuchtung der vordersten LED-Lampe in der ersten Spalte und Zeile, müssen einige Kriterien erfüllt sein. Da heißt es wird positive Spannung an die erste Zeile und 0V zur ersten Spalte zugeführt. Dennoch ist für die gesamte Zeile die Spannung verfügbar (ABB3). Da aber nur die erste Spalte Strom erhält, bleiben die restlichen dunkel und transferieren keinen Strom zum GND. Die Realisierung dieser Arbeit zeigt unter anderem textuelle als auch graphische Darstellungen auf der Matrix, die sich „bewegen“ und einen sinnvollen Gehalt ergeben. Dadurch müssen mehrere LEDs gleichzeitig leuchten und ferner müssen hierdurch auch mehrere Spalten und Zeilen aktiviert sein. Die Problematik an dieser Stelle ist jedoch, dass Spalten und Zeilen einer Matrix nicht gleichzeitig⁵⁰ angesteuert werden können. Hinsichtlich einer erfolgreichen Realisierung, gibt es aber eine Methoden mit der dieses Problem umgangen werden kann bzw. es so aussehen lässt, als ob die LEDs tatsächlich gleichzeitig oder aber „bewegt“ arbeiten. Für diesen Zweck eignet sich das bereits erwähnte Multiplexing. Mit Hilfe dieser Technik verwandelt sich die Matrix zu einem Display. Im Grunde genommen werden dafür die Zeilen der Matrix der Reihe nach schnell an-und ausgeschaltet. Angesichts dessen leuchtet also jeweils immer nur eine Zeile auf einmal. Obgleich das menschliche Auge diese Bewegung nicht nachverfolgen kann, so dass die Wahrnehmung ein anderes Bild vermittelt⁵¹.

⁴⁸ Vgl. ebd.

⁴⁹ Vgl. Richter, Peter

⁵⁰ Vgl. Dransfeld, Damon

⁵¹ Vgl. TenDance

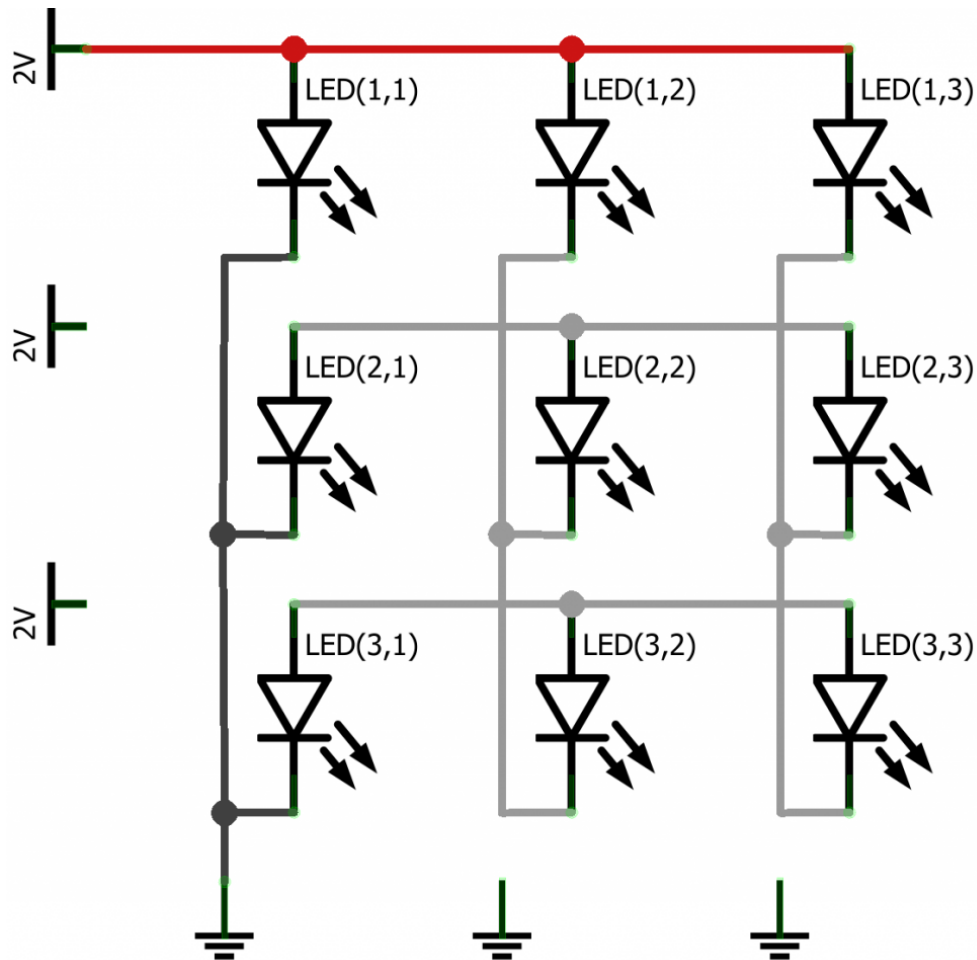


Abbildung 3 LED-Matrix

Solch ein Effekt ist ab 30Hz⁵², was einem 30-maligen Aufleuchten in einer Sekunde entspricht, möglich. Zum Vergleich aus Kapitel 2.2 besitzt der Raspberry Pi 700Mhz. Daher ist das für den Einplatinencomputer mehr als nur ein leichtes Unterfangen.

2.5. Verfügbare Distributionsvarianten

Neben den Anschlussmöglichkeiten, gewährt der Raspberry Pi auch diverse Auswahlmöglichkeiten in der Distributionsauswahl⁵³. Diese Eigenschaft wird durch das Booten des Betriebssystems über eine SD- bzw. Micro-SD-Karte ermöglicht. Dazu muss lediglich eine Karte mit einem Betriebssystemabbild bespielt und in den Lese-Slot eingesteckt werden. Für welches der Systeme man sich entscheidet hängt letztendlich vom Verwendungszweck ab. Die Großzahl der Distributionen ist auf Basis von Debian

⁵² Vgl. ebd.

⁵³ Vgl. Dragicevic, Marko

GNU/Linux⁵⁴ entwickelt worden. Darunter auch das Betriebssystem Raspbian⁵⁵, welches eine speziell optimierte Version für den Raspberry Pi darstellt und sich besonders großer Beliebtheit⁵⁶ in der Online-Community erfreut. Zudem liefert diese Distribution einige wichtige Programme für den Start. Nichtsdestotrotz ist die Grundstruktur für den Raspberry Pi Open Source. Das heißt, im Zuge von Modifikationen⁵⁷ haben sich aus den verschiedenen Communities auch diverse Betriebssysteme entwickelt und kursieren nun im Netz. Gleichwohl ist auch das Interesse namhafter Akteure für den kleinen Einplatinencomputer erheblich gestiegen, so dass beispielsweise Android⁵⁸ und Microsoft⁵⁹ eigene angepasste Versionen haben. Jedes Betriebssystem bietet Vor- und Nachteile und kann jederzeit durch ein anderes ersetzt werden. Für die vorliegende Arbeit, wird ausschließlich das Betriebssystem Raspbian Wheezy eingesetzt. Dennoch werden in der nachfolgenden Tabelle (TAB2) einige wichtige Distributionssysteme vorgestellt die von Raspberry Pi unterstützt werden.

⁵⁴ Vgl. Britz, Christian

⁵⁵ Vgl. Raspberry Pi Foundation

⁵⁶ Vgl. Biglesp

⁵⁷ Vgl. Steuer, Richard

⁵⁸ Vgl. Thompson, John

⁵⁹ Vgl. Westaway, Luke

Distributions-name	Besonderheiten	Zweckeignung
Raspbian	Standard BS, simpler Einstieg	Speziell für die Hardware des Raspberry Pi, Allrounder
NOOBS	Sammlung verschiedener Distributionen	Erlaubt den Zugriff auf mehrere Betriebssysteme
Pidora	Basis Fedora Linux	Desktop, Server
Arch Linux ARM	Stets aktuelle Software, Fortgeschrittene	Büro, Multimedia, Spiele
OpenElec	Media Center	Filme, Videos, Musik
Ubuntu Mate	Ressourcensparend	Vereinfachte Desktop-Ubuntu-Version
Kali Linux	Arsenal zum Ausmachen von Sicherheitslücken	IT Forensik, Penetrationstests
Windows 10	erlaubt Open-Source Sprachen	Embedded Devices

Tabelle 2 Distributionsübersicht

2.6. Vergleichseinschätzung zu anderen Einplatinencomputer

Der Abschnitt wird in Kürze die Vorteile des Raspberry Pi aufzeigen und dann in einer Tabelle eine kleine Vergleichsgegenüberstellung (TAB3) zu anderen Einplatinencomputern dar geben. Neben der Größe und der stromsparenden Eigenschaft, ist auch der Preis⁶⁰ ein ausschlaggebender Grund zum Raspberry Pi zu greifen. Das Besondere am Modell B+ ist ein eingebauter Schaltregler⁶¹, welcher eine stabilere Versorgung erlaubt und nicht unter Volllast der USB-Ports zusammenbricht. Dank des verbauten Chips ist der Raspberry Pi Modell B+ von der Leistung her so stark wie Smartphones vor etwa 4-5 Jahren⁶². Damit kann der kleine Einplatinencomputer zwar nicht mit den neuesten High-End-Computern mithalten, aber dafür ist er auch nicht ausgelegt. Dennoch ist eine Nutzung als Multimediagerät möglich. Dies ist mit nur wenig Aufwand zu erreichen. Generell ist gerade für Anfänger der Raspberry Pi

⁶⁰ Vgl. Raspberry Pi Foundation

⁶¹ Vgl. Thoma, Jörg

⁶² Vgl. T. Nick

interessant, da sie mit Leichtigkeit Projekte umsetzen können. Das setzt sich zum einen aus der Open-Source Mentalität in der Software⁶³ und aus den Anschaffungskosten zusammen. Der finanzielle Rahmen liegt immer in einem überschaubaren Feld. Abgesehen davon sind die GPIO-Pins eine hervorragende Möglichkeit, um diverse Geräte anzusteuern, die mit einem normalen Computer nicht ohne weiteres in Betracht kommen würden. Diverse andere Einplatinencomputer sind im Grunde genommen fähig die Leistung des Raspberry zu übersteigen. Jedoch unterscheidet sich dann der Anschaffungspreis oder das Einstiegslevel. Dieser Aspekt darf aber nicht vernachlässigt werden. Denn die Realisierung soll sich auch an Privatpersonen und Studenten richten. Deswegen spielt das Einstiegsniveau eine tragende Rolle für einen Projektaufbau. Was nicht außer Acht gelassen werden darf, ist, dass es möglich ist den Raspberry Pi in einem Projekt zu integrieren und dieses dann kommerziell zu nutzen. Dabei müssen entsprechende Kennzeichnungen für die Nutzung eingeschlossen sein⁶⁴. Dennoch ist es möglich mit einer Idee, die auf dem Einplatinencomputer basiert, Kapital zu erwirtschaften.

⁶³ <http://www.golem.de/news/miniplatine-ist-das-raspberry-pi-wirklich-offene-hardware-1403-104564.html>

⁶⁴ <https://www.raspberrypi.org/blog/starting-a-business-with-a-raspberry-pi/>

Bewertungs-Kriterien	Raspberry Pi Modell B+	<i>Cubieboard</i>	<i>ODROID-X2</i>	Beagle Bone Black Rev. C	Arduino Uno REV 3
Leistung /Takt	Broadcom BCM 2835 mit 700MHz	Allwinner A10 mit 1 GHz	Cortex-A9 MPCore Quad Core mit 1,7 GHz	Sitara AM3358 mit 1GHz	Atmel SAM3X8E mit 84 MHz
Betriebs-system	Linux-Distributionen, Windows 10	Linux-Dist., Android 4.X	Ubuntu, Android 4.X	Linux-Dist., Android 4.2	Windows, Linux, Mac OSXF
GPIO	26	96	40	69	14
Programmier-barkeit	C/C++, Python, Java, Scratch, Perl, Bash uvm.	C	C/C++, Java	JavaScript	C/C++, AVR-Assembler
Kosten	Ca. 35€	Ca. 89€	Ca. 135€	Ca. 50€	Ca. 20€
Einstiegs-level	Einsteiger & Fort-geschritten	Fort-geschritten	Einsteiger & Fort-geschritten	Einsteiger & Fort-geschritten	Einsteiger
Zweck	Multimedia, Steuerung GPIO-Pins	Steuerung GPIO-Pins	Multimedia, Steuerung GPIO-Pins	Multimedia, Steuerung GPIO-Pins	Steuerung GPIO-Pins

Tabelle 3 Vergleichaufstellung Einplatinencomputer

3. Vorbereitung und Installation

Das nächste Kapitel wird auf den Vorbereitungsprozess vor Inbetriebnahme des Raspberry Pi eingehen. Neben dem nötigen Zubehör, wird auch der Anschluss zwischen den einzelnen Komponenten beschrieben. Dies beinhaltet zudem die Installation der Distribution auf der Micro-SD-Karte.

3.1. Hardwarekomponenten

Zur erfolgreichen Realisierung dieses Projektes müssen neben dem Raspberry Pi und der LED-Matrix selbst weitere Komponenten miteinbezogen werden. Um ein Anschließen der LED-Matrix mit den GPIO-Pins des Raspberry Pi zu erreichen, sind sogenannte Jumper Kabel als Steckverbindung zwischen diesen zwei Elementen von Nöten. Wichtig ist, dass sowohl weibliche (ABB5) als auch männliche (ABB4) Stecker im Equipment zu finden sind.



Abbildung 4 Jumper Kabel männlich



Abbildung 5 Jumper Kabel weiblich

Genauso elementar sind die sogenannten Flachbandkabel (ABB6). Sie stellen die Verbindung zwischen den einzelnen LED-Matrizen her. Hinzu kommt ein 2,1mm Buchsen Stromadapter mit einer Schraubklemme (ABB7).



Abbildung 6 Flachbandkabel



Abbildung 7 Buchsen Stromadapter

Ein weiterer wichtiger Bestandteil ist auch die Micro-SD-Karte (ABB8). Diese sollte zumindest eine Kapazität von 8GB und eine Schreibgeschwindigkeit von zwei Megabyte je Sekunde beherbergen. Die Annahme zu dieser Behauptung resultiert aus der Speichereinnahme⁶⁵ der Distribution Raspbian mit 4 GB, wodurch nur geringe Kapazitäten für eigene Daten übrig bleiben. An dieser Stelle empfiehlt sich natürlich eine höhere Schreibgeschwindigkeitsrate, um Einschränkungen bei den Zugriffszeiten zu beseitigen. Die nächste Komponente, die zur erfolgreichen Realisierung des Projektes verwendet werden muss ist das Netzteil (ABB9). Im Prinzip spielt hier nur die Stromspannung eine Rolle. Daher könnte auch ein Handyladegerät als Stromquelle dienen. Es muss lediglich eine konstante Spannung von 2 oder mehr Ampere bestehen. Sollte die Leistung darunter liegen, können Komplikationen bei den Lese- und Schreibvorgängen eintreten⁶⁶. Abgesehen davon können sich hierdurch Fehlfunktionen im Raspberry Pi einschleichen oder gar das ganze System einfrieren. Eine Art Absicherung⁶⁷ besteht durch die im 5V Eingang des Raspberry Pi verbaute Polyfuse bei 2 Ampere. Sind jedoch alle vier USB Ports in Verwendung, ist eine höhere Stromleistung vorausgesetzt. Dies ist auch der Fall, wenn beispielsweise externe Festplatten angeschlossen werden. Daher empfiehlt es sich bei stromfressenden Komponenten noch ein USB-Hub dazwischenschalten. Hierdurch lässt sich die Gefahr vor Beschädigung ausschließen⁶⁸.



Abbildung 8 Micro-SD-Karte



Abbildung 9 Netzteil

⁶⁵ Vgl. Elektronik-Kompodium.de

⁶⁶ Vgl. SASCHA

⁶⁷ Vgl. Henderson, Gordon

⁶⁸ Vgl. SASCHA

Relevant ist in diesem Zusammenhang auch die Videoausgabe des Computers. Je nachdem wie modern der eingesetzte Display ist, kommt ein HDMI/DVI Adapterkabel oder ein handelsübliches HDMI-Kabel für die Ausgabe in Frage (ABB11). Der Vorteil vom HDMI-Kabel ist die gleichzeitige Audio- und Videoausgabe. Dieser Aspekt ist besonders interessant, wenn der Raspberry Pi für Multimediazwecke eingesetzt wird. Andernfalls muss die Audioausgabe zusätzlich über ein 3,5mm Klinkenanschluss und Audioboxen hergestellt werden. Damit der Raspberry Pi Zugriff zum Internet erhält ist ein Ethernet-Kabel (ABB10) elementar. Zur Eingabesteuerung dürfen USB-Tastatur und Maus nicht fehlen. An dieser Stelle wird eine Kombination aus beiden eingesetzt (ABB11). Damit nun vollends ein reibungsloser Start gewährleistet werden kann, muss für die Konfiguration der Micro-SD-Karte ein vorhandener Rechner mit entsprechendem Lesegerät zur Verfügung stehen. Sind alle benötigten Ressourcen betriebsbereit, kann die Verbindung zwischen den einzelnen Komponenten vollzogen werden.



Abbildung 10 Ethernet-Kabel



Abbildung 11 HDMI-Kabel



Abbildung 12 Rii i8 Tastatur

3.2. Aufbau und Verbindung der LED-Matrix mit dem Raspberry Pi

Um Beschädigungen durch Überspannung o.ä. auszuschließen, sollte die Hardware bis zum Vollenden der korrekten Verbindung nicht an das Stromnetz geschlossen werden. Dies betrifft sowohl den Raspberry Pi selbst, als auch die LED-Matrix. Erfahrungsgemäß versucht eine eingeschaltete LED-Matrix Strom von den GPIO Pins zu ziehen.

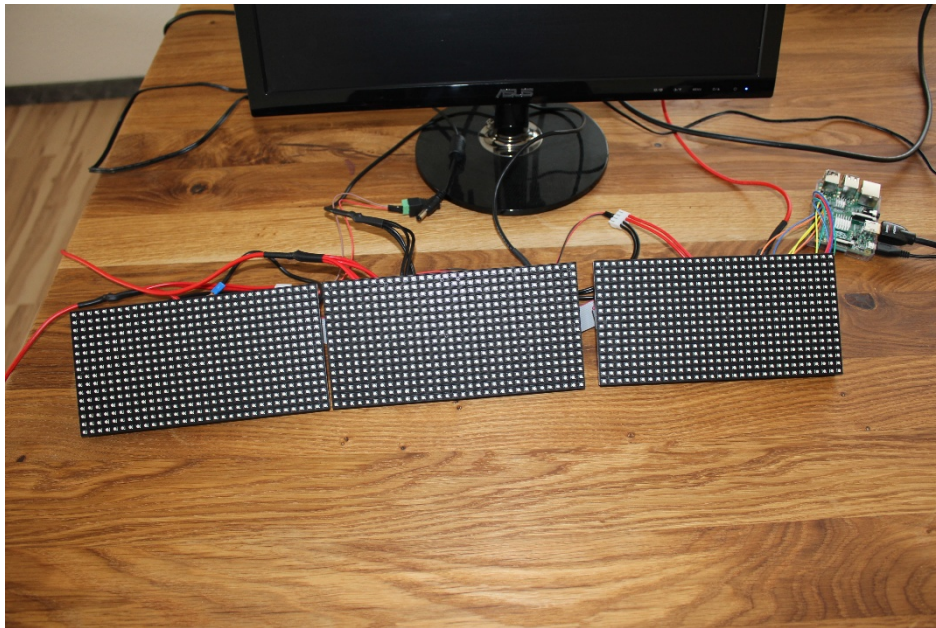


Abbildung 13 Angeschlossene LED-Matrizen

Wie bereits erwähnt, können hierdurch, besonders bei fehlerhafter Verkabelung, Schäden am Einplatinencomputer entstehen. Daher ist es ratsam, alle Geräte erst an das Stromnetz anzuschließen, wenn die Verbindung korrekt vollzogen worden ist. Da für das vorgesehene Projekt insgesamt drei LED-Module (Abb13) in Reihe geschaltet werden, ist die Verbindung dieser Elemente auch ein Teil des Aufbaus. Hierfür sind die bereits erwähnten Flachbandkabel, in vorstehendem Fall 16-polig, für den angegebenen Zweck sinngemäß. Auf dem LED-Modul befindet sich ein am linken und am rechten Rand (ABB16) ein Pin Steckplatz (ABB14) für das eben angesprochene Verbindungskabel. Auf diesem befindet sich am Kopf eine Orientierungseinfärbung. Die jeweiligen „Köpfe“ (ABB15) sollten auf dem LED-Modul gegenüberliegend jeweils bei „Input“ und „Output“ eingesteckt werden.

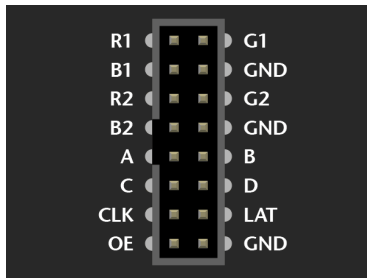


Abbildung 14 Steckplatz Matrix

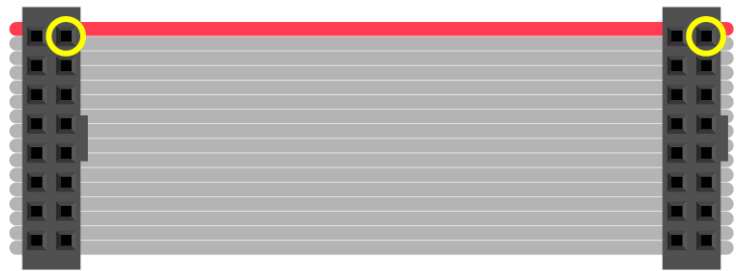


Abbildung 15 Kopfseite

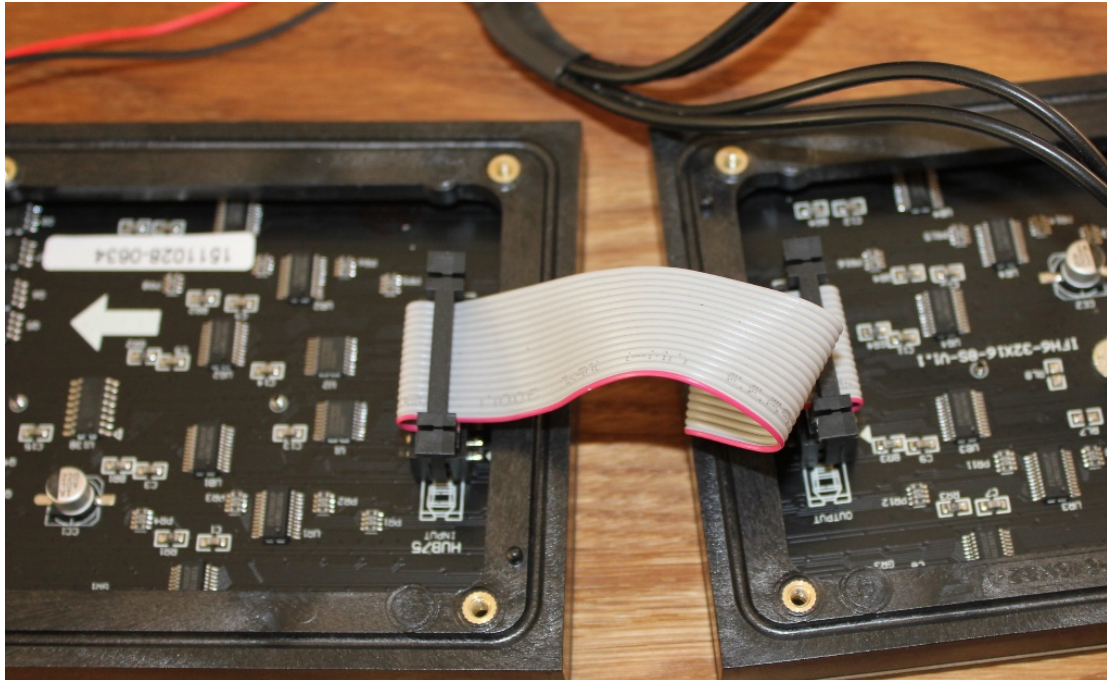


Abbildung 16 Anschluss zwischen Matrizen

Nun folgt das Anschließen der Jumper Kabel zwischen der GPIO-Schnittstelle und dem nahestehendem LED-Modul. Dies sollte wie auf der unten stehenden Abbildung (ABB17) vollzogen werden. Dabei spielt die Farbe der Jumper Kabel keine wesentliche Rolle. Lediglich die korrekte Verbindung vom Einplatinencomputer zur LED-Matrix muss stimmen.

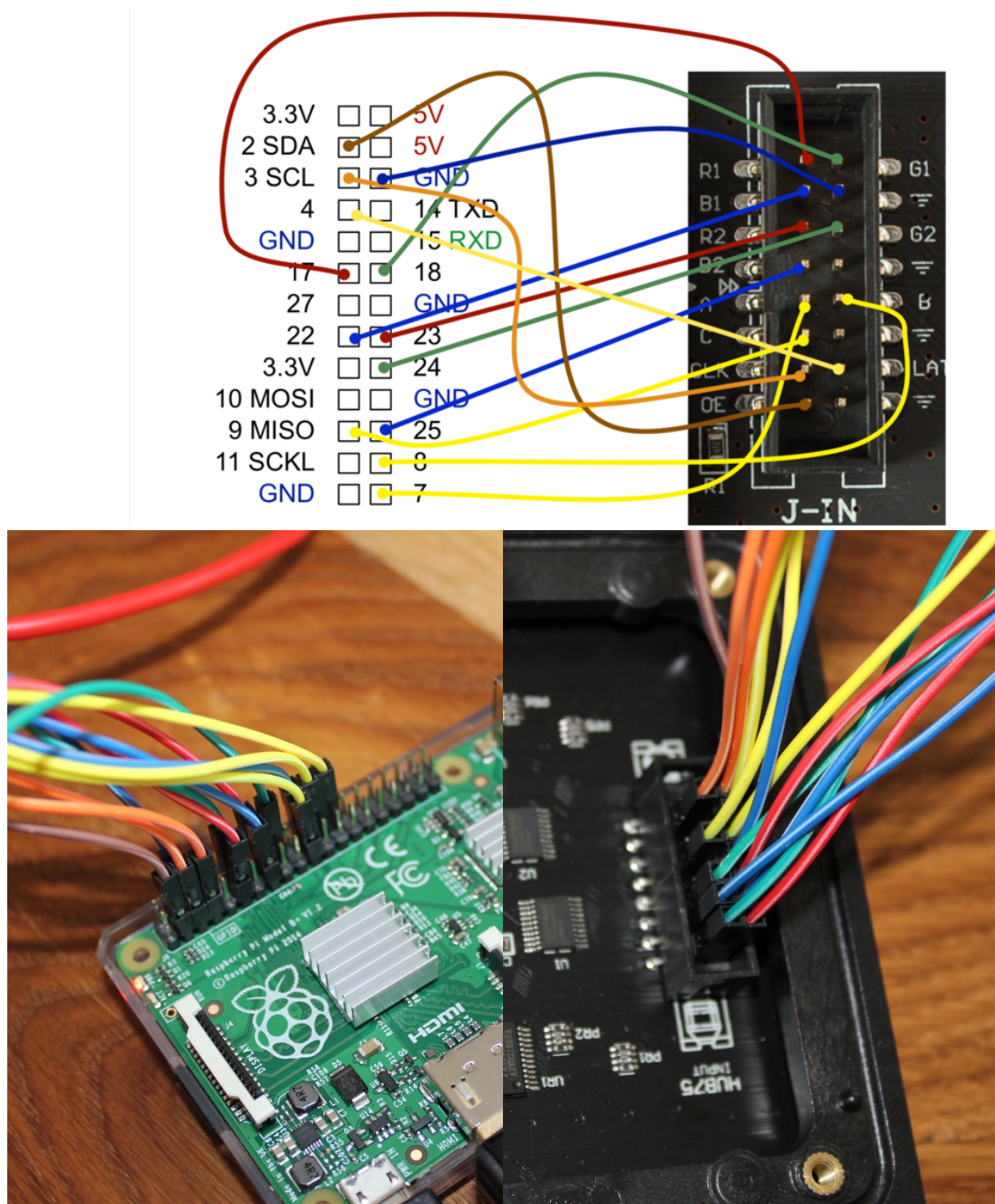
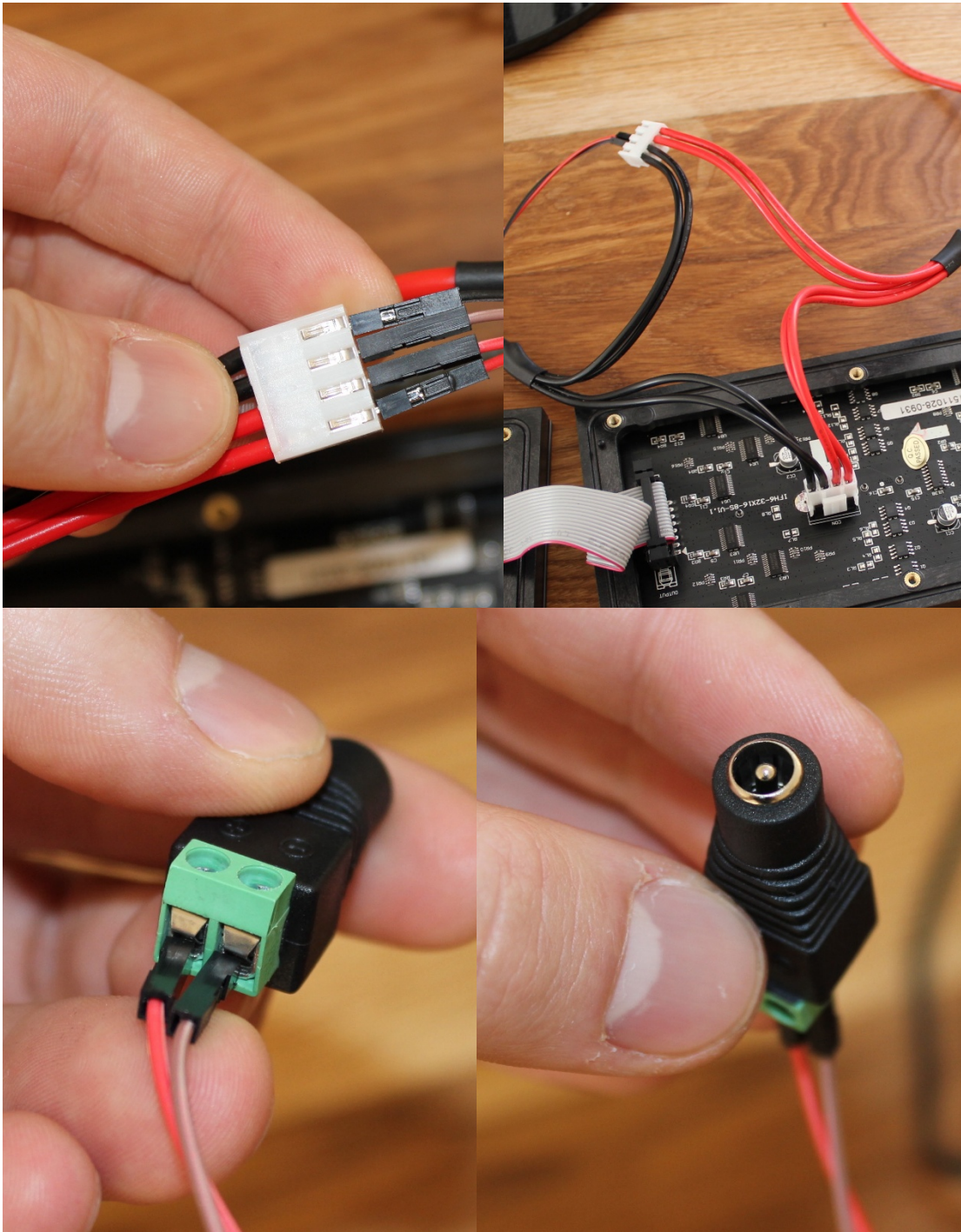


Abbildung 17 GPIO Verkabelung

Nach Abschluss und erneuter Überprüfung der Verkabelung, muss als nächstes Strom für die LED-Matrizen hergeschafft werden. Dazu werden 4-polige Stromverbindungskabel auf die Rückseite der LED-Matrizen unter der Aufschrift **CON** angeschlossen. Im Normalfall befindet sich direkt darunter eine Pfeilmarkierung, die auf den Anschluss hinweist. Mit Jumper Kabeln kann dann unter Verwendung des jeweiligen (+)- und (-)-Pols eine Verbindung zu jeder Matrix geschaffen werden. Eins der Verbindungskabel führt zu dem Buchsen Steckadapter. Die Brücke zu den beiden sind zwei Jumper Kabel, welche entsprechend des Pols in die Schraubklemme des Buchsenadapters eingefügt werden. Auch wenn die Stromkabel sowohl zwei positive

als auch negative Leitungen besitzen, reicht jeweils nur eine Verbindung zum Adapter. Dabei spielt es keine Rolle welcher der beiden Leitungen besetzt wird. Es muss aber sichergestellt werden, dass die positive Leitung zum roten und die negative zum schwarzen Kabel führen. Sind alle Schritte soweit erfolgt, sieht das Ergebnis in etwa wie auf der (ABB18) aus. Die LED-Matrizen werden an dieser Stelle jedoch noch nicht an ein Netzteil angesteckt.



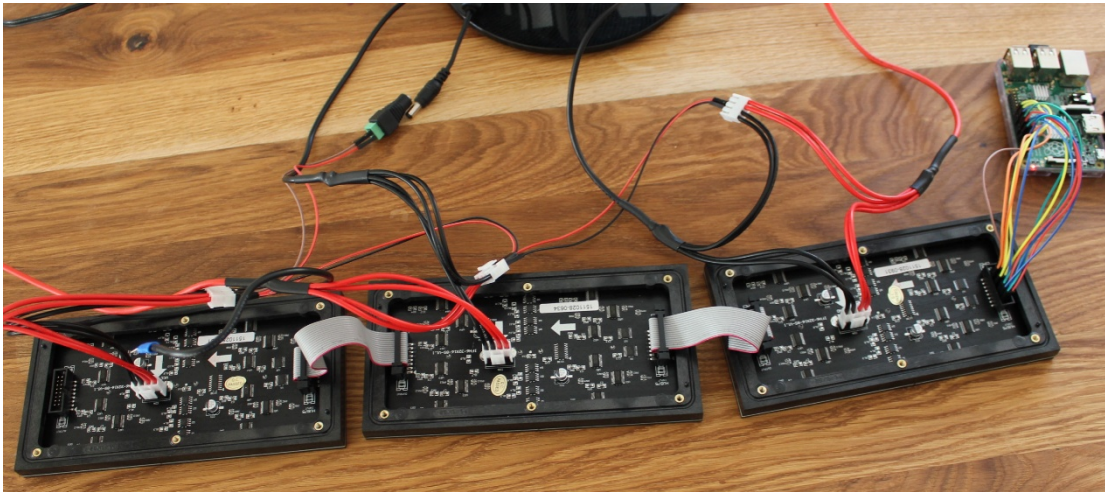


Abbildung 18 Verkabelungsablauf LED-Matrizen

3.3. Imagevorbereitung

Die in dieser Arbeit verwendete Distribution für den Raspberry Pi ist Raspbian, welches vom Entwickler nicht nur an die Architektur des Einplatinencomputer angepasst wurde, sondern auch die Installation von Softwarepaketen erlaubt, die im weiteren Verlauf nützlich sein können⁶⁹. Eine Methode ein Image, also ein Abbild eines Betriebssystems auf den Raspberry Pi zu bringen, wurde bereits angesprochen. Es handelt sich dabei, um die Transferierung auf eine SD-Karte bzw. Micro-SD-Karte. Hierbei muss die Kopie bitweise erfolgen. Diese Vorgehensweise erweist sich als schnell und unkompliziert. Ebenso lässt sich mit dieser Methode, vom zu installierendem Image ein Backup erstellen. Das spart Zeit und erleichtert eine Wiederherstellung des Systems. Dazu muss anfänglich das Image über einen netzwerkfähigen Computer heruntergeladen werden. Als Quelle eignet sich hierfür die Webseite⁷⁰ der Raspberry Gründer. Dabei ist frei wählbar ob dieser Vorgang über eine ZIP Datei oder per Torrent⁷¹ geschieht. An dieser Stelle geeignet ist jedoch die ZIP Datei⁷². Hierauf befindet sich jeweils immer ein aktuelles und stabil laufendes Image. Dies sollte nun mit einem entsprechenden Tool entpackt werden können. Das Image lässt sich nun an der Endung „img“ erkennen. Diese Datei enthält eine bitweise Kopie der Distribution. Um nun dieses Image lauffähig zum Raspberry Pi zu bringen ist entweder eine unbenutzte oder eine formatierte SD-Karte notwendig. Ersteres lässt die Formatierung entfallen und erlaubt im nachfolgenden Schritt die Überschreibung.

⁶⁹ Vgl. Raspberry Pi Foundation

⁷⁰ Vgl. Raspberry Pi Foundation

⁷¹ Vgl. Sharpened Productions

⁷² Vgl. Raspberry Pi Foundation

Hierbei legt das System zwei Partitionen an, unterteilt in FAT32 und EXT-Version⁷³. Andernfalls muss erst eine Löschung des Inhalts stattfinden, um das ursprüngliche Format wiederherzustellen. Nun je nachdem welches Betriebssystem an dieser Stelle genutzt wird, unterscheidet sich der folgende Ablauf in seinen Einzelheiten. Um eine fehlerfrei Formatierung durchzuführen, sollte ein entsprechendes Formatierungstool eingesetzt werden. Solche Tools sind im Netz kostenlos⁷⁴ erhältlich und eignen sich meistens besser dafür als die distributionseigenen Formatierungsassistenten. Ist die SD-Karte angeschlossen, sollte einem Formatierungsvorgang nichts mehr im Wege stehen. Dazu muss man im Normalfall den Datenträger vorher auswählen und unter Optionen die Formatierung auswählen. Sind alle Einstellungen abgeschlossen, erfolgt eine Übersicht der tatsächlichen Größe auf der SD-Karte. Um das Image letztendlich noch auf der Karte zu kopieren, wird ein weiteres Tool gebraucht. Dieses muss nämlich die SD-Karte bootfähig machen und dazu verwendet wird der SD Formatter⁷⁵, welcher sich auch kostenlos im Internet herunterladen lässt. Ist das Programm installiert und funktionsbereit, muss in einem Auswahlkontext (Drive) die SD-Karte gewählt werden. (ABB19)

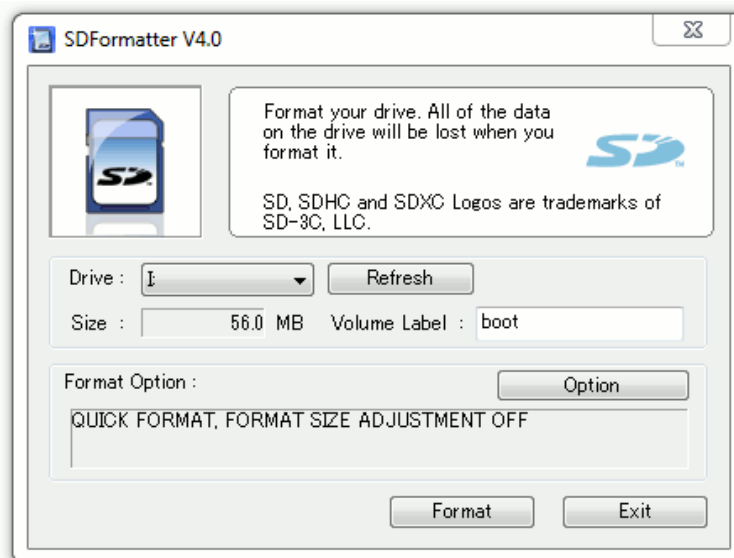


Abbildung 19 SD-Formatter 1

⁷³ Vgl. Hoffman, Chris

⁷⁴ Vgl. Schanze, Robert

⁷⁵ Vgl. SD Association

Im nächsten Schritt wird das vorher entpackte Image Raspbian Wheezy geöffnet und mit einem Klick auf „write“ der Überschreibungsprozess gestartet. Sobald der Vorgang beendet ist, kann die Micro-SD-Karte letztendlich ausgeworfen und aus dem entsprechenden Lesegerät entfernt werden. An dieser Stelle ist es sinnvoll ein Backup bzw. Wiederherstellungspunkt zu erstellen, bevor die SD-Karte mit dem Einplatinencomputer operieren kann⁷⁶. Eben erwähntes lässt sich mit dem Tool Win32 Disk Imager bewerkstelligen. Dafür muss die SD-Karte wieder an das Lesegerät eines lauffähigen Computers mit dem genannten Programm zur Verfügung stehen. Nach Starten des Tools, ist in dem Dialogfeld, welches vorher zur Imageauswahl gedient hat, ein eigener und nicht vorhandener Dateiname mit der Endung „img“ zu erstellen. Die Schaltfläche „Öffnen“ führt nun zum Fenster, an dessen Stelle wir den Schreibvorgang vollführt haben. Doch an dieser Stelle, ist die Option „read“ für ein Lesevorgang der vorhandenen Daten auf der SD-Karte auszuführen. Dieser Vorgang ist dann äußerst nützlich, wenn die Daten auf einen größeren Wechseldatenträger übertragen und wiederhergestellt werden sollen. Im Vorfeld sollte auch diese Micro-SD-Karte wie oben beschrieben, formatiert und entsprechen bootfähig gemacht werden, bevor das Backup, mit dem eigenen Namen im Dialogfenster ausgewählt und auf die Karte geschrieben wird. Ist der Datenträger nun in einer fertigen Variante vorhanden, kann er in das Lesegerät des Raspberry Pi gesteckt werden. (ABB20)

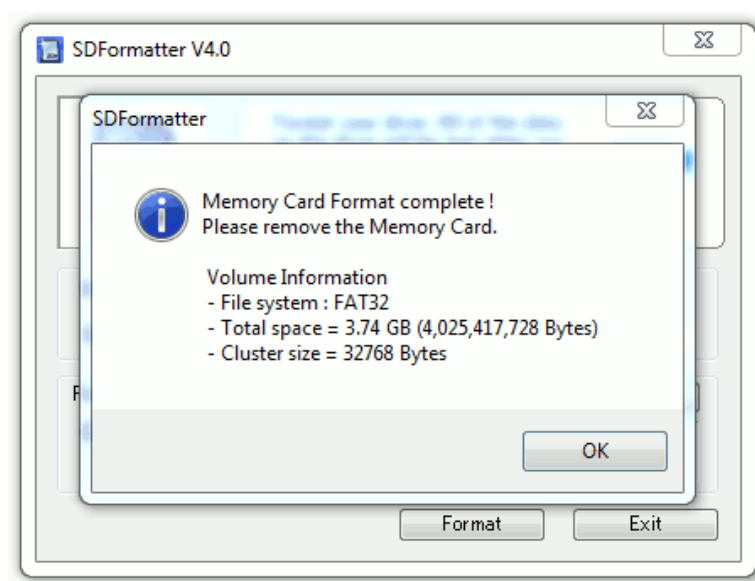


Abbildung 20 SD Formatter 2

⁷⁶ Vgl. Raspberry Pi Foundation

4. Konfigurationsschritte

Die Handlung des folgenden Kapitels, knüpft an das vorhergehende und setzt eine korrekte Verbindung aller Hardware-Komponenten sowie die Vorbereitung der Distribution voraus. Das heißt sämtliche Schritte, die nun nach dem ersten Start des Raspberry Pi unternommen werden müssen, sind hier näher erläutert. Darunter fallen unter anderem die Einstellung für das Tastaturlayout, sowie Zeit- und Spracheinstellungen.

4.1. Starten des Bootvorgangs

Vorausgesetzt, die Distribution, in diesem Fall Raspbian, wurde korrekt auf die SD-Karte installiert, erfolgt nun eine grundlegende Konfiguration für den Einsatz beim Nutzer. Optimal ist an dieser Stelle der direkte Anschluss eines Bildschirms, einer Maus und Tastatur am Einplatinencomputer selbst. Kann aus irgendeinem Grund kein Bildschirm direkt mit dem Raspberry Pi verbunden werden, besteht die Option über das Netzwerk unter Verwendung eines sogenannten SSH Client⁷⁷ auf den Computer zuzugreifen. Diese erlaubt einen verschlüsselten Fernzugriff auf den Raspberry Pi mithilfe eines anderen Computers. Passend dazu gibt es ein Tool namens PuTTY⁷⁸, welches einen solchen Zugriff über einen Nutzernamen und einem Passwort erlaubt. Da zu diesem Zeitpunkt noch keine Veränderungen des Systems unternommen worden sind, lauten die Standardwerte⁷⁹ für die Verbindung beim Benutzer „pi“ und beim Passwort „raspberry“. Bei der Eingabe ist zu beachten, dass die Tastaturbelegung noch nicht vorgenommen wurde und daher einige Buchstaben wie beispielsweise „Z“ und „Y“ vertauscht sind.

4.1.1. Vorgehensweise bei der raspi-config

Sind alle vorhergehenden Schritte korrekt durchgeführt worden und es liegt eine direkte Verbindung zu einem Bildschirm vor, bootet der Raspberry Pi und es erscheint das Konfigurationsfenster raspi-config⁸⁰. Dieses äußert sich in Form eines Blue Screens mit einer grauen Box, welche verschiedene Optionen, bereithält (ABB21). Ähnlich wie

⁷⁷ Vgl. ITWissen.info

⁷⁸ Vgl. Tatham, Simon

⁷⁹ Vgl. Raspberry Pi Foundation

⁸⁰ Vgl. Raspberry Pi Foundation

das BIOS bei einem handelsüblichen PC⁸¹. Zweck dieser Optionen ist es allgemeine Grundeinstellungen am System vorzunehmen. Um sich nun in diesem Optionsmenü fortzubewegen, sind die Pfeiltasten unten und oben zu verwenden. Hierdurch bewegt sich der markierte Bereich auf die entsprechende Option und signalisiert die nächste Auswahlmöglichkeit. Mithilfe der Enter-Taste lässt sich das jeweilige Menü betreten und mit der ESC-Taste verlassen. Beim Drücken auf die rechte Pfeiltaste springt der markierte Bereich auf die zwei unteren Tasten „Select“ und „Finish“. Wird an dieser Stelle nochmals die linke Pfeiltaste betätigt, springt der markierte Bereich auf die Optionstexte zurück. Ist stattdessen die Bewegung zwischen den zwei unteren Tasten beabsichtigt, ist dies durch den Tabulator zu bewerkstelligen. Hinzuzufügen bleibt, dass bei langläufigen Optionsmenüs auch das Betätigen eines Buchstaben zu einer Auswahl führen kann. Beispielsweise ist dann der Buchstabe „G“ für Germany im Zusammenhang mit der Zeitzoneneinstellung zu drücken. Das erspart unnötiges Scrollen. Sollte dies dennoch nötig sein, kommen hierfür die „Bild hoch“ und „Bild runter“ Tasten in Frage. Die erste Option namens 1 „Expand Filesystem“ ist interessant, da das Image in Eigenregime auf die Micro-SD-Karte geschrieben worden ist. Der Raspberry Pi nutzt also nur etwas mehr als 3GB Speicher durch das Image. Beim Wählen dieser Option kann sich die Installation des Systems auf die gesamte Karte verteilen und erlaubt somit weiteren Platz für Ordner und Dateien auf dem Einplatinencomputer. Wichtig beim Betreten dieser Option ist, dass es keine Bestätigung über eine Änderung gibt und der Raspberry Pi nochmals gebootet werden muss um das Ganze verfügbar zu machen. Mit Hilfe der nächsten Option 2 „Change User Password“ lassen sich Standardbenutzername und Passwort ändern.

⁸¹ Vgl. Munteanu, Llie-Alexandru

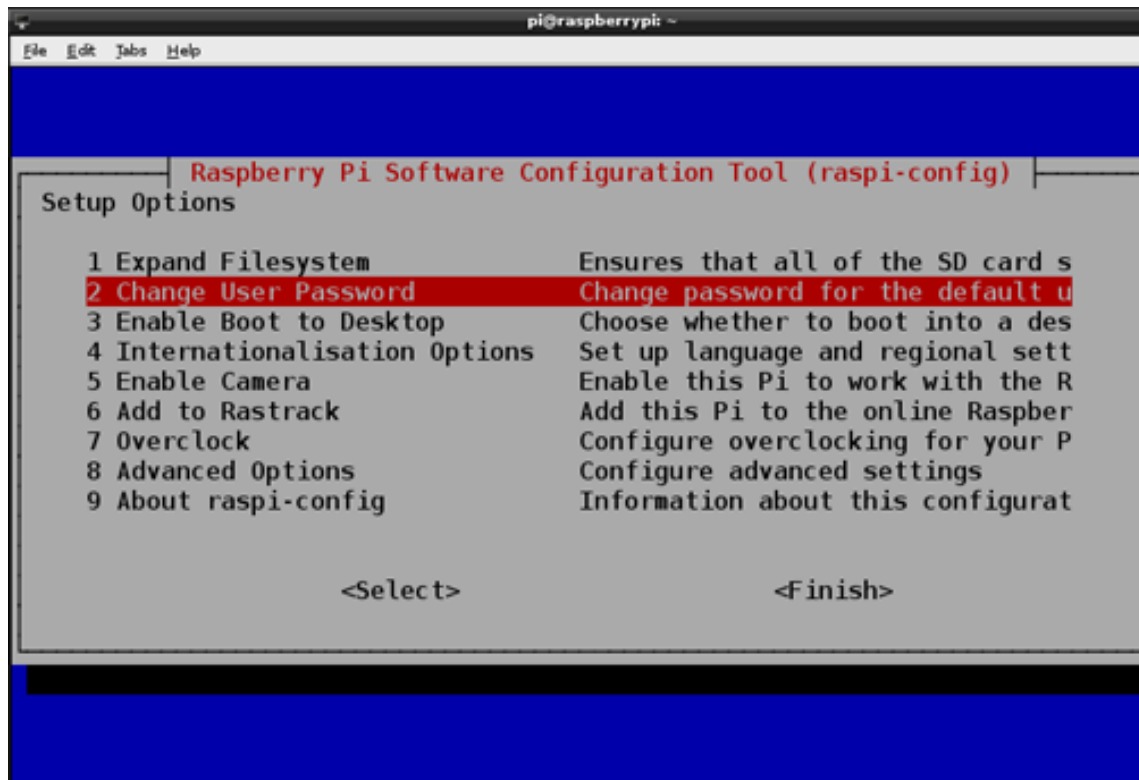
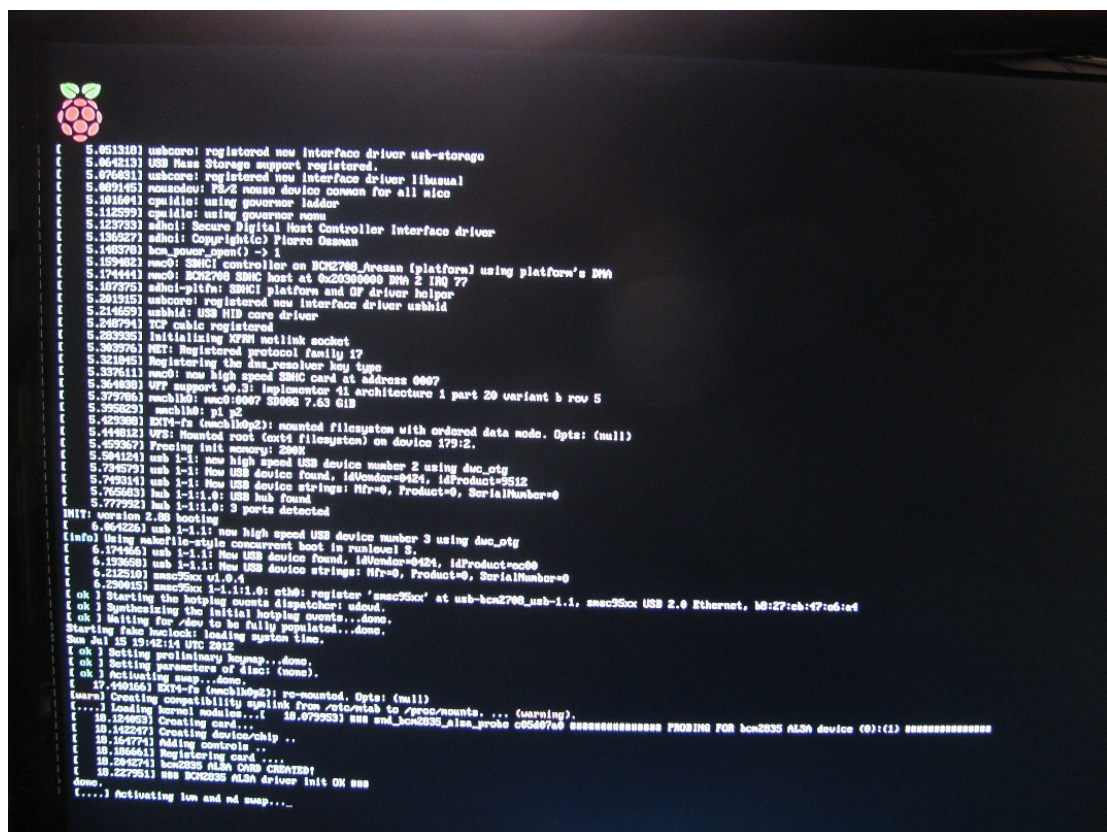


Abbildung 21 Raspi-config

Der Aspekt der Sicherheit, sollte genügen, um diese Werte möglichst umzuschreiben. Anschließend kommt die Option 3 „Enable Boot to Desktop/Scratch“ an die Reihe. Hierbei wird lediglich entschieden was der Raspberry Pi beim Booten macht. Beispielsweise kann man nach dem Booten direkt die Kommandozeile laden oder stattdessen den Desktop als Präferenz einstellen. Damit der Raspberry Pi am Einsatzort Deutschland sinngemäß verwendet werden kann ist ein Handeln unter dem Punkt 4 „Internationalisation Options“ Pflicht. Das System kann hierbei sowohl Standort als auch Sprachoptionen vollziehen. Damit die sachgerechte Nutzung eintritt, sollte für den Standort (*change locale*) „de_DE.UTF-8“ als auch für den (*default locale*) ausgewählt werden. Das System wird für die Kreierung der Parameter einige Sekunden benötigen. Immer noch im selben Menü lässt sich auch der geografische Zeitwert umstellen. In diesem Fall empfiehlt es sich Europa und Berlin anzugeben. Über das NTP lädt sich der Einplatinencomputer die Informationen zur Zeitangabe und stellt sofort die Uhr um. Daraufhin ist eine Anpassung des Tastaturlayouts auf die deutsche Sprache unter (*Change Keyboard Layout*) das nächste Ziel. Veränderungen am Tastaturlayout werden im Normalfall sofort übernommen, jedoch erfordert dies ein erneutes Booten des Systems. Mithilfe der Option 5 „Enable Camera“ lässt sich ein

Kameramodul unterstützen. Diese Einstellung ist nicht zwingend erforderlich, kann aber zu jeder Zeit geändert werden. Option 6 „*Rastrack*“ zeigt lediglich eine Art Heatmap⁸² der Raspberry Pi Geräte weltweit auf und erfüllt in diesem Projekt keine weitere Funktion. Hiernach ist unter Option 7 „*Overlock*“ die Übertaktung des Raspberry Pis möglich. Dadurch kann die Leistung des Einplatinencomputers weiter erhöht werden, jedoch ist auch diese Option nicht im Interesse dieser Arbeit. Daher ist nur noch die Option 8 „*Advanced Options*“ ansprechend. An dieser Stelle können unter anderem die Werte für die Grafikkarte verändert und der Netzwerk-Gerätenamen des Einplatinencomputers umbenannt werden. Eine Belassung des Standardnamens kann beim Anschluss mehrerer Computer zur Verwirrung führen. Beim Abschließen aller Änderungen kann im Konfigurationsfenster des raspi-config nun mit „*Finish*“ eine Bestätigung erfolgen.



```

[ 5.051310] usbcore: registered new interface driver usb-storage
[ 5.064213] USB Mass Storage support registered.
[ 5.076481] usbcore: registered new interface driver libusual
[ 5.091451] nouveau: PM/2 nousso device common for all nico
[ 5.101641] cpuidle: using governor ladder
[ 5.112599] cpuidle: using governor none
[ 5.123733] sdhci: Secure Digital Host Controller Interface driver
[ 5.136327] sdhci: Copyright(c) Pierre Ossman
[ 5.140378] bcm_pwrseq: bcm_pwrseq() -> 1
[ 5.159482] mmc0: SDHCI controller on BCM2706_0 (platform) using platform's DMA
[ 5.174441] mmc0: BCM2706 SDHC host at 0x20300000 DMA 2 IRQ 77
[ 5.187375] sdhci-pltfm: SDHCI platform and OF driver helper
[ 5.201315] usbcore: registered new interface driver usbhid
[ 5.214639] usbhid: USB HID core driver
[ 5.248791] TCP cable registered
[ 5.253935] Initializing NFSM netlink socket
[ 5.263766] NET: Registered protocol family 17
[ 5.321945] Registering the dma_resourcer key type
[ 5.337611] mmc0: new high speed SDHC card at address 0007
[ 5.364638] VFP support v0.3: implementor 41 architecture 1 part 20 variant 6 rev 5
[ 5.377086] mmcblk0: mmc0:0007 SD000 7.63 GiB
[ 5.392629] mmcblk0: p1 p2
[ 5.423388] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
[ 5.448112] VFS: Mounted root (ext4 filesystem) on device 179:2.
[ 5.453527] Freeing init memory: 200K
[ 5.501241] usb 1-1: new high speed USB device number 2 using dwc_otg
[ 5.749314] usb 1-1: New USB device found, idVendor=0424, idProduct=5512
[ 5.765683] hub 1-1:1.0: USB hub found
[ 5.772922] usb 1-1:1.0: 3 ports detected
INIT: version 2.88 booting
[ 6.064226] usb 1-1.1: new high speed USB device number 3 using dwc_otg
[ 6.174766] Using ramfs-style concurrent boot in ramfs! S.
[ 6.193658] usb 1-1.1: New USB device found, idVendor=0424, idProduct=ec00
[ 6.212510] smac550x v1.0.4
[ 6.260415] smac550x 1-1.1.1.0: eth0: register 'smac550x' at usb-bcm2706-usb-1.1, smac550x USB 2.0 Ethernet, 18:27:eb:47:a6:e4
[ ok ] Starting the booting system dispatcher: udevd.
[ ok ] Synthesizing the initial booting events...done.
Starting the booting: loading system time.
Sun Jul 15 15:42:14 UTC 2012
[ ok ] Setting preliminary hostname...done.
[ ok ] Setting parameters of drc: (none).
[ ok ] Activating swap...done.
[ 17.450166] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[ warn] Creating compatibility symlink from /etc/mtab to /proc/mounts, ... (warning).
[ 18.124603] Creating caud... 18.077953] sbs and_bcm2835_alma_probe 006497a0 0000000000000000 PRObing for bcm2835 alsa device (0):(1) 0000000000000000
[ 18.167774] Adding device/chip ..
[ 18.186661] Registering card...
[ 18.204241] bcm2835 ALSA CARD CREATED!
[ 18.227951] sbs bcm2835 alsa driver init OK sbs
done.
[....] Activating lvm and md swap....

```

Abbildung 22 Erster Bootvorgang

⁸² Vgl. Raspberry Pi Foundation

Hiermit ist das Konfigurationsfenster beendet und es wird nach einem erneuten Bootvorgang gefragt (ABB22). Auch dies wird bestätigt und schließlich findet man sich auf dem grafischen Interface (ABB23) wieder.

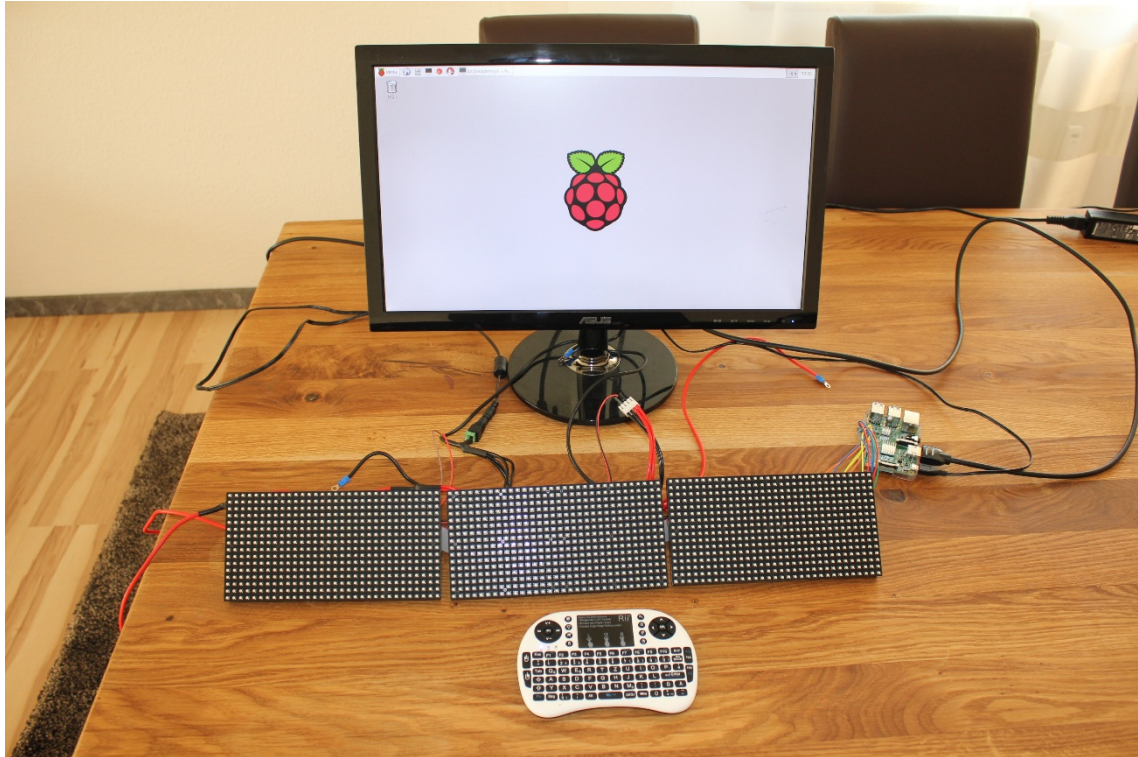


Abbildung 23 Übersicht Komponenten

Ist zu diesem Zeitpunkt noch eine Konfiguration in der raspi-config vorzunehmen, kann über das Terminal⁸³ (ABB24) mit dem untenstehenden Befehl das Fenster aufgerufen werden.

```
pi@raspberrypi ~ $ sudo raspi-config
```

Sind zu Beginn keine entsprechenden Peripheriegeräte wie Bildschirm, Maus und Tastatur vorhanden oder wird ein SSH Zugriff vorgenommen, muss das Konfigurationsfenster auf diese Weise aufgerufen werden.

⁸³ Vgl. Raspberry Pi Foundation

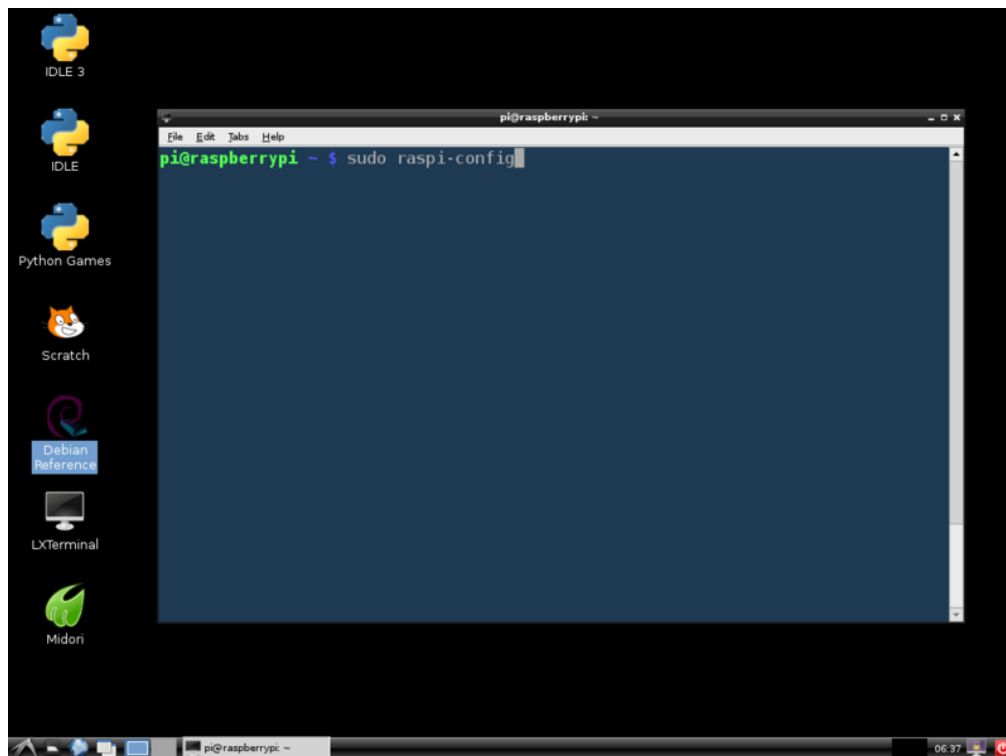


Abbildung 24 Terminal

4.1.2. Funktionsweise Konsolenbefehl *sudo*

Im Kapitel zuvor ist der Befehl *sudo* zum Einsatz gekommen. Hinter diesen vier Buchstaben verbergen sich die Zugriffsrechte des Root-Kontos⁸⁴. Dieser Benutzer darf sämtliche Vorgänge ohne Nachfrage ausführen und macht den Systemadministrator-Account überflüssig. Da mit diesem Befehl im Prinzip die Löschung des gesamten Betriebssystems möglich ist, sollte die Ausführung mit Vorsicht gehandhabt werden. Im Normalfall jedoch fragt das System nach einem Passwort zur Überprüfung der Authentizität des aufrufenden Benutzers. Gegebenenfalls können die berechtigten Benutzer im Verzeichnis⁸⁵ */etc/sudoers* eingesehen und verändert werden. Die Vorgehensweise mit dem Befehl *sudo* ist in einem Handbuch, in den sogenannten manpages (ABB25), näher beschrieben.

⁸⁴ Vgl. Kampert, Daniel

⁸⁵ Vgl.ebd.

```

pi@raspberrypi: ~/ledmatrix/rpi-rgb-led-matrix
Datei Bearbeiten Reiter Hilfe
SUDO(8)                                MAINTENANCE COMMANDS                                SUDO(8)

NAME
  sudo, sudoedit - execute a command as another user

SYNOPSIS
  sudo -h | -K | -k | -V

  sudo -v [-AknS] [-g group name|#gid] [-p prompt] [-u user name|#uid]

  sudo -l[l] [-AknS] [-g group name|#gid] [-p prompt] [-U user name]
  [-u user name|#uid] [command]

  sudo [-AbEHnPS] [-C fd] [-g group name|#gid] [-p prompt] [-r role]
  [-t type] [-u user name|#uid] [VAR=value] [-i | -s] [command]

  sudoedit [-AnS] [-C fd] [-g group name|#gid] [-p prompt]
  [-u user name|#uid] file ...

DESCRIPTION
  sudo allows a permitted user to execute a command as the superuser or
  another user, as specified by the security policy. The real and
  effective uid and gid are set to match those of the target user, as
  Manual page sudo(8) line 1 (press h for help or q to quit)

```

Abbildung 25 Manual Page

4.1.3. Fernzugriff über SSH

Der Raspberry Pi ist bis hierher nur direkt über die Konsole angesprochen worden. Angenommen kein Monitor ist zum Anschließen bereit oder der Benutzer möchte aus der Ferne die Steuerung übernehmen. Dann eignet sich der Fernzugriff über die sogenannte *Secure Shell* (SSH) hervorragend für diesen Prozess. Solcher kann je nach Betriebssystem unterschiedliche Vorbereitungen erfordern. Da für die Realisierung sowohl über das Betriebssystem Windows als auch MAC OS eine SSH-Verbindung hergestellt wird, werden auch beide Varianten näher beleuchtet. Die Software, die für die Verbindung zu trage kommt, ist bereits auf dem hier verwendeten Betriebssystem Raspbian Wheezy installiert. Auf dem Windows-Rechner hingegen muss das Programm PuTTY⁸⁶ aus dem Internet heruntergeladen werden. Ist der Vorgang beendet, ist das Starten des Programms ohne Installation möglich. Daraufhin öffnet sich ein Fenster (ABB26) und die erste Inbetriebnahme kann erfolgen. Mit dem Programm PuTTY sind auch andere Verbindungsmöglichkeiten denkbar, aber an dieser Stelle nicht relevant. In den Grundeinstellungen, ist SSH als Verbindungstyp zu wählen sowie die IP-Adresse

⁸⁶ Vgl. Tatham, Simon

des Raspberry Pi und der Port 22. Der Port ist ein Standardwert⁸⁷ für die Kommunikation mit SSH, daher bleibt dieser unverändert.

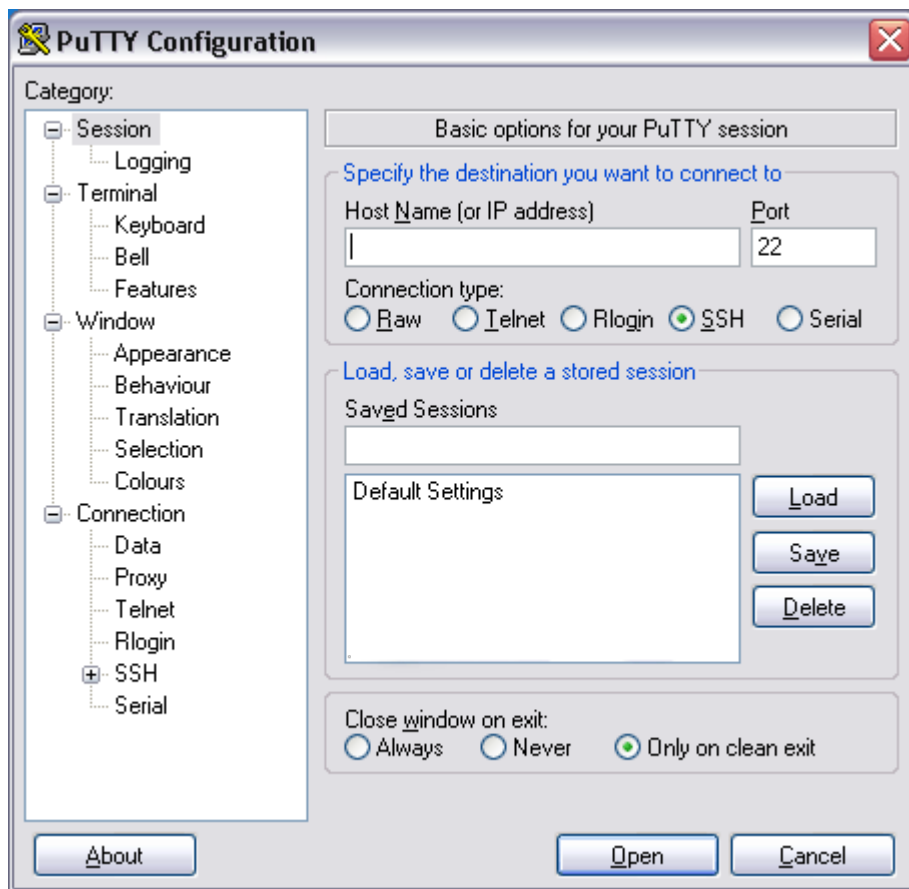


Abbildung 26 PuTTY Konfiguration

Darunter befindet sich ein Feld mit Namen *Saved Sessions*. Diese dient zur Abspeicherung der eben eingegebenen Daten. Das heißt mit einem Verbindungsnamen, den diese Sitzung erhält, werden die Verbindungseinstellungen behalten und bei der nächsten Verwendung übernommen. Ist ein Name gefunden, kann auf die Schaltfläche *Save* geklickt werden und er erscheint in der darunter stehenden Liste. Mit *Load* ist die vorkonfigurierte Einstellung jederzeit abrufbar. Nach der Einrichtung, führt ein Doppelklick auf den Konfigurationsnamen oder die Schaltfläche *open* zum Start der Verbindung. Ist innerhalb einer gewissen Zeit keine Reaktion zu vernehmen wird eine Benachrichtigung mit folgender Meldung erscheinen: *Network error: Connetion timed out*. Das deutet meistens darauf hin, dass die IP-Adresse nicht korrekt ist. Nach Überprüfung und erfolgreicher Verbindung, erscheint seitens PuTTY eine Warnung (ABB27) zur Vertrauenswürdigkeit des Servers

⁸⁷ Vgl. Assigned Numbers Authority

bzw. dem Raspberry Pi. Diese Meldung erscheint lediglich bei erstmaliger Verwendung. Nach Bestätigung dieser Meldung, wird der Raspberry als vertrauenswürdige Quelle eingestuft. Daraufhin wird sich ein Konsolenfenster öffnen und die übliche Abfrage des Benutzers und seinem Passwort für den Raspberry Pi einleiten. Nach erfolgreicher Eingabe, kann die Konsole wie beim Raspberry Pi genutzt werden.



Abbildung 27 PuTTY Warnung

Im Übrigen wird diese Methode auch *headless-Konfiguratio*⁸⁸ genannt, da sie sozusagen ohne eigenen Kopf bzw. Monitor erfolgt. Auf dem Mac OS sieht der Vorgang ein wenig anders aus. Durchaus einfach wird das Ganze mit einer zusätzlichen Anwendung (ABB28) namens Raspberry Pi Finder⁸⁹. Der Einplatinencomputer sowie der fernsteuernde Computer müssen im Netzwerk verbunden sein. Der Pi Finder hat nach wenigen Sekunden den Einplatinencomputer gefunden. In der Eingabemaske werden Benutzername und Passwort eingegeben, so dass die Verifizierung in einem Gang erledigt wird. Letztendlich kann über die Schaltfläche Terminal das übliche Konsolenfenster des Raspberry Pi geöffnet werden.

⁸⁸ Vgl. MaxEmbedded

⁸⁹ Vgl. GitHub, Inc.

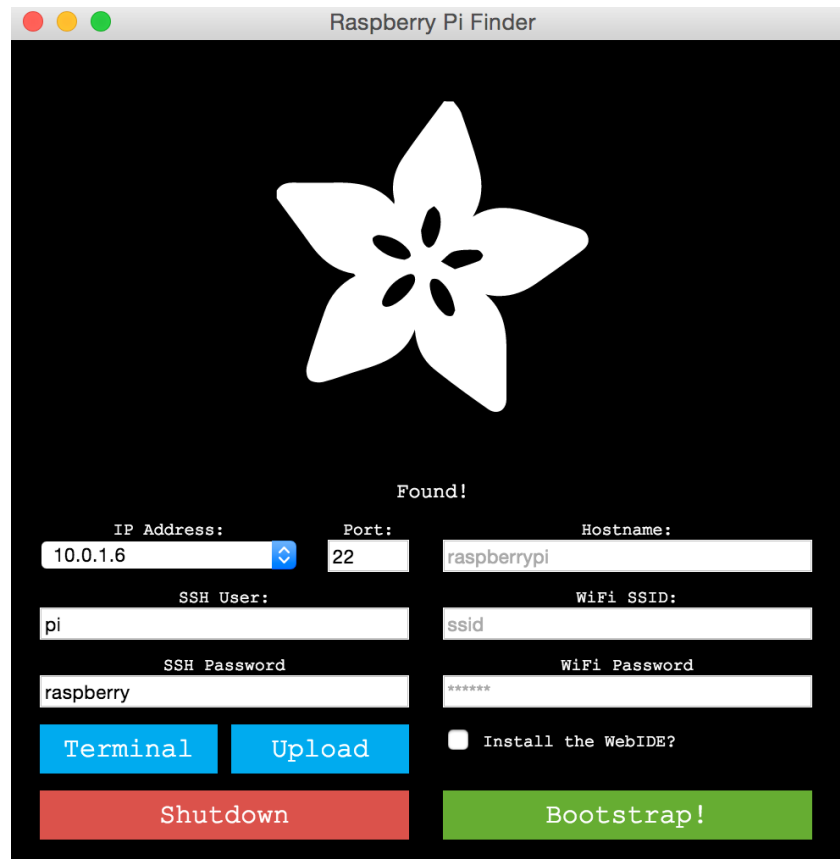


Abbildung 28 Raspberry Pi Finder

4.2. Update

Da alle nötigen Grundeinstellungen, sowie eine Verbindung zum Netzwerk besteht, muss für den weiteren Ablauf dieses Projektes ein Update durchgeführt werden. Dafür vorgesehen und verwendet wird der Paketverwaltungssystem *APT*⁹⁰, mit dessen Hilfe eine Aktualisierung der Firmware und Software stattfindet.

4.3. Raspberry Pi-update

Die Firmware des Raspberry Pi sollte regelmäßig auf den neuesten Stand gebracht werden, um Vorteile aus den Verbesserungen und Problembeseitigungen zu erhalten. Dadurch können eventuell auch Kompatibilitätsschwierigkeiten mit diversen Komponenten beseitigt und Projekte weiterentwickelt werden. Über das *LXTerminal*⁹¹, welches als Kommandozeile dient, kann man die jeweils aktuelle Kernelversion der Firmware herausfinden. Dafür eignet sich der unten stehende Befehl welcher umgehend die eine Ausgabe wiedergibt.

⁹⁰ Vgl. Savie, Ravi

⁹¹ Vgl. Raspberry Pi Foundation

```
pi@raspberrypi ~ $ uname -a
```

Die Ausgabe, sollte der hier unten stehenden in etwa ähneln. Um genauer zu verstehen, was die Angaben bedeuten, muss der Text aufgespalten werden. Hierbei ist die erste Zahlenfolge als Kernel Version zu verstehen, welche am 18 Oktober 2014 kompiliert worden ist. Die Nummer hinter dem Hash ist die Versionsnummer des Kernels.

```
Linux raspberrypi 3.12.25+ #691 PREEMPT Thu Oct 18 19:42:02 BST 2014 armv6l  
GNU/Linux
```

Zusätzlich kann auf diese Weise noch die Version der GPU Firmware ermittelt werden, um auch hier ein entsprechendes Update bereitzustellen. Dazu ist nachstehende Zeile in diese Form einzugeben.

```
pi@raspberrypi ~ $ /opt/vc/bin/vcgencmd
```

Ist die Eingabe getätigt gibt das System die angefragten Informationen heraus und stellt sie wie unten angegeben dar.

```
Oct 5 2014 23:28:10  
Copyright (c) 2012 Broadcom  
version 342141 (release)
```

Sind nun alle wichtigen Informationen bekannt, erfolgt die Aktualisierung auf die neueste Version mit einem kurzen Befehl. Das Besondere an dieser Zeile jedoch ist das

```
pi@raspberrypi ~ $ sudo rpi-update
```

Kommando *sudo*, hinter dem sich sogenannte Root-Rechte verbergen. Oder auch anders Ausgedrückt, die Rolle des Administrators wird zugeteilt und führt tiefgreifende Veränderungen im Betriebssystem durch.

```

pi@raspberrypi ~ $ sudo rpi-update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** Performing self-update
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 7813 100 7813    0     0  1376      0  0:00:05  0:00:05 --:--:-- 9864
*** Relaunching after update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** We're running for the first time
*** Backing up files (this will take a few minutes)
*** Backing up firmware
*** Backing up modules 3.12.28+
*** Downloading specific firmware revision (this will take a few minutes)
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 168    0  168    0     0   115      0  --:--:--  0:00:01 --:--:--  171
100 23.0M 100 23.0M    0     0  274k      0  0:01:25  0:01:25 --:--:-- 138k
*** Updating firmware
*** Updating kernel modules
*** depmod 3.12.33+
*** Updating VideoCore libraries
*** Using HardFP libraries
*** Updating SDK
*** Running ldconfig
*** Storing current firmware revision
*** Deleting downloaded files
*** Syncing changes to disk
*** If no errors appeared, your firmware was successfully updated to 5ba37078ca
6bea96e02ceb94f9f77e95c4621e45
*** A reboot is needed to activate the new firmware
pi@raspberrypi ~ $

```

Abbildung 29 Rpi-update

Der Raspberry Pi muss nach Ausgabe des oben stehenden Bildverlaufs (ABB29) neugestartet werden, um die Installation des Updates zu beenden. In dieser Beziehung kann jetzt entweder rechts unten auf dem Desktop, mit Hilfe des Abmelden- und Shutdown Button fortgefahren werden, oder aber einer der unten stehende Befehle wird im Terminal eingegeben.

```
pi@raspberrypi ~ $ sudo shutdown -r now
```

```
pi@raspberrypi ~ $ sudo shutdown -h now
```

Hiermit ist der Updatevorgang abgeschlossen, wodurch die Vorbereitung der Software als nächstes auf den Plan rückt.

4.4. Paketverwaltungssystem apt

Zwecks der erfolgreichen Installation von Software wird in diesem Kontext noch ein weiteres Werkzeug gebraucht. Die Rede ist vom *APT Paketverwaltungssystem*⁹²,

⁹² Vgl. Savie, Ravi

welches unter der Distribution Raspbian operiert. Dieses ist in der Lage, Programmpakete zu suchen, zu installieren, zu aktualisieren und letztendlich zu Verfügung zu stellen. Eben solche Pakete können unter anderem in Software, Libraries und diversen anderen Funktionalitätsabhängigkeiten vorkommen. Damit dieses Verwaltungstool die entsprechend aktuellen Softwareversionen für den Nutzer bereitstellen kann, muss die Eingabe des untenstehende Codes in das Terminal erfolgen. Auf diese Weise wird jetzt ein Software Update durchgeführt.

```
pi@raspberrypi ~ $ sudo apt-get update
```

Das APT erfragt, nach erfolgreicher Überprüfung und Berechnung der benötigten Ressourcen, den Nutzer nach einer Bestätigung. Die folgende Abbildung (ABB30) zeigt eine Ausgabe dieser Art auf. Bejaht man die Frage, wird das Update ausgeführt und mit einem Neustart sämtliche Änderungen übernommen.

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  comerr-dev javascript-common krb5-multidev libavahi-common-dev libdbus-1-dev
  libgcrypt11-dev libgnutls-dev libgnutls-openssl27 libgnutlsxx27 libgpg-error-dev
  libgssrpc4 libidn11-dev libkadm5clnt-mit8 libkadm5srv-mit8 libkdb5-6 libkrb5-dev
  libldap2-dev libp11-kit-dev librtmp-dev libssh2-1-dev libssl-doc libtasn1-3-dev
  wwwconfig-common
Suggested packages:
  doc-base apache2 httpd krb5-doc libasound2-doc libcurl3-dbg libgcrypt11-doc
  gnutls26-doc krb5-user mysql-client postgresql-client
The following NEW packages will be installed:
  comerr-dev javascript-common krb5-multidev libasound2-dev libavahi-client-dev
  libavahi-common-dev libcurl4-openssl-dev libdbus-1-dev libexpat1-dev libgcrypt11-dev
  libgnutls-dev libgnutls-openssl27 libgnutlsxx27 libgpg-error-dev libgssrpc4
  libidn11-dev libjs-jquery libkadm5clnt-mit8 libkadm5srv-mit8 libkdb5-6 libkrb5-dev
  libldap2-dev libopus-dev libp11-kit-dev librtmp-dev libssh2-1-dev libssl-dev
  libssl-doc libtasn1-3-dev libudev-dev wwwconfig-common
0 upgraded, 31 newly installed, 0 to remove and 4 not upgraded.
Need to get 9,023 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Abort.
pi@raspberrypi / $
```

Abbildung 30 Apt-get-update

5. Einrichtung Software und Programmcode

Ein weiteres Fortsetzen des Projektes erfordert die Erfüllung aller oben beschriebenen Schritte. An dieser Stelle wird jedoch nur eine einzelne LED-Lampe mit Widerstand (ABB31) angeschlossen, um jeweils die Programmierung und die GPIO-Pins zu testen. Des Weiteren werden hierdurch die Schnittstellen zur Software vorbereitet und die generelle Ansteuerung der GPIO-Pins zu externen Geräten aufgezeigt. Für den soeben beschriebenen Prozess eignet sich ein sogenanntes Framework⁹³, also ein Programmgerüst, das die Schnittstelle zwischen der Hardware und den Libraries für die verschiedenen Programmiersprachen darstellt. Für diesen Einsatz ist sowohl das Programm wiringPi⁹⁴ als auch die Ansteuerung per Bash-Befehl erwägbar.

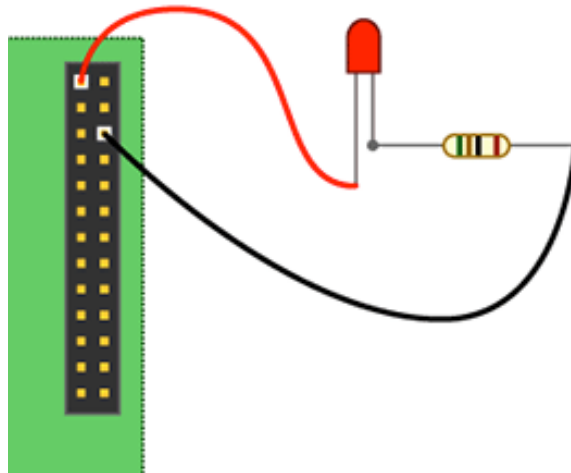


Abbildung 31 Test-LED

5.1. Export Bash

Die erste Variante ist anhand eines sogenannten Exports⁹⁵ via Bash-Befehlen ausführbar. Um GPIO-Pins ansteuern zu können, müssen diese bei jedem Neustart des Systems freigeschaltet werden. Die Pins sind auf dem Einplatinencomputer als Datei ausgedrückt und unter `/sys/class/gpio` auffindbar. Davon ausgehend Pin 7 der Schnittstell soll angesprochen werden, welcher laut der Abbildung (ABB2). aus Kapitel 2.1, GPIO4 entspricht, sodann wird dem System übermittelt diesen Pin zu exportieren. Die Problematik an dieser Stelle, ist jedoch das Zugriffsrecht für diese Datei. Denn

⁹³ Vgl. Henderson, Gordon

⁹⁴ Vgl. ebd.

⁹⁵ Vgl. Hammerschmit, Anton

unter Linux hat normalerweise nur das Root-Konto⁹⁶ Rechte, um an Konfigurations- und Kontrolldateien etwas zu verändern. Dementsprechend wird das System das Voranschreiten verhindern.

```
pi@raspberrypi ~ $ sudo echo 7 >/sys/class/gpio/export  
bash: export: Permission denied
```

Um an den Informationsaustausch zwischen Hard- und Software dranzukommen, ist entweder der Wechsel zum Root-Konto oder die Umschreibung der Rechtevergabe im Export-Verzeichnis möglich⁹⁷. Da diese Eingabe mit jedem Neustart wiederholt wird, ist die zweite Option vorteilhafter. Hierfür eignet sich das Kommando *chmod* in Verbindung mit dem Befehl *sudo*, wodurch sich die Attribute⁹⁸ für die Berechtigung ändern lassen.

```
pi@raspberrypi ~ $ sudo chmod 222 /sys/class/gpio/export  
/sys/class/gpio/unexport
```

Auf diese Weise erzeugt der Kernel⁹⁹ das vorher beschriebene Verzeichnis */sys/class/gpio/gpio23* und erlaubt die Arbeit mit der hiesigen Datei. Der letzte Schritt vor der Ansteuerung selbst ist die Definition der Aufgabe des GPIO. Insofern wird damit entweder ein Lesen *in* oder ein Schreiben *out* gekennzeichnet.

```
pi@raspberrypi ~ $ echo "out" > /sys/class/gpio/gpio4/direction
```

Demnach ist der Pin jetzt bereit zur Ansteuerung. Wichtig diesbezüglich ist auch, dass der GPIO-Pin zwei unterschiedliche Zustände annehmen kann. Aufgrund dessen lässt

⁹⁶ Vgl. ebd.

⁹⁷ Vgl. Kampert, Daniel

⁹⁸ Vgl. Torvalds, Charles

⁹⁹ Vgl. Kampert, Daniel

sich die Stromzufuhr ein- oder ausschalten. Gemäß dem Binärsystem¹⁰⁰ stellt die „1“ den angeschalteten Zustand und in diesem Fall eine Spannung von 3,3 Volt dar. Wohingegen die „0“ das Fließen von Strom unterbindet. Da es zunächst um eine Einschaltung geht, muss der Code wie folgt aussehen.

```
pi@raspberrypi ~ $ echo "1" > /sys/class/gpio/gpio4/value
```

Fällt nun die Entscheidung, die eben beschriebene Exportierung rückgängig zu machen, weil zum Beispiel ein anderer Pin angesteuert werden soll. So ist dies mit der nachstehenden Zeile möglich.

```
pi@raspberrypi ~ $ echo 4 > /sys/class/gpio/unexport
```

5.2. Export via wiringpi2

Das im Kapitelanfang erwähnte Programm wiringPi erlaubt einerseits eine unkomplizierte Steuerung der GPIO-Schnittstelle und zum anderen die Einbindung von Sprachen wie beispielsweise C, C++ und Python¹⁰¹. Ebenso werden auf diese Weise Funktionen mitgeliefert, die es erlauben Daten aus den Pins, die man als Eingangs- bzw. Leseschnittstelle verwendet, auszuwerten¹⁰². Diese zuletzt genannte Eigenschaft, wird im späteren Verlauf der Arbeit noch eine entsprechende Rolle zugute geschrieben. Dieses nützliche Programm ist jedoch nicht standardmäßig auf dem Raspberry Pi installiert. Über ein sogenanntes GIT¹⁰³, ein Versionsverwaltungstool, ist diese Prozedur realisierbar.

Damit ein reibungsloser Installationsprozess durchgeführt werden kann, sollte erstmals die Aktualität des Raspberry Pi sichergestellt sein und dann das eben beschriebene GIT hinzugezogen werden.

```
pi@raspberrypi ~ $ sudo apt-get update && sudo apt-get upgrade  
pi@raspberrypi ~ $ sudo apt-get install git-core
```

¹⁰⁰ Vgl. Elektronik-Kompendium.de

¹⁰¹ Vgl. Henderson, Gordon

¹⁰² Vgl. ebd.

¹⁰³ Vgl. Torvalds, Linus

Über GIT kann darauffolgend der Download von wiringPi stattfinden und sodann die Installation erfolgen. Wobei auch die Navigation zum entsprechenden Ordner zwischengeschoben werden muss. Damit ist ein gelungener Abschluss der Installation möglich.

```
pi@raspberrypi ~ $ git clone git://git.drogon.net/wiringPi
pi@raspberrypi ~ $ cd wiringPi
pi@raspberrypi ~ /wiringPi $ ./build
```

Um die heruntergeladenen Version im weiteren Verlauf aktuell zu halten ist die Benutzung des anschließenden Befehls zu berücksichtigen. Eine Überprüfung des Status der Raspberry Pi Ports sollte erfolgen, um eventuelle Fehler bei der Eingabe ausschließen zu können. Daher können Informationen und die Pinnbelegung des Einplatinencomputers mittels der unten stehenden Codes eingesehen werden.

```
pi@raspberrypi ~ /wiringPi $ gpio -v
pi@raspberrypi ~ /wiringPi $ gpio readall
```

Es erscheint ein Resultat, das der unten angegebenen Abbildung (ABB32) gleicht und dem Modell Raspberry Pi Modell B+ entspricht.

```
pi@raspberrypi:~/wiringPi $ gpio readall
```

BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1	2		5v		
2	8	SDA.1	IN	1	3	4		5V		
3	9	SCL.1	IN	1	5	6		0v		
4	7	GPIO. 7	IN	1	7	8	1	ALT0	TxD	15
		0v			9	10	1	ALT0	RxD	16
17	0	GPIO. 0	OUT	0	11	12	0	IN	GPIO. 1	1
27	2	GPIO. 2	IN	0	13	14		0v		18
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4
		3.3v			17	18	0	IN	GPIO. 5	5
10	12	MOSI	IN	0	19	20		0v		24
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6
11	14	SCLK	IN	0	23	24	1	IN	CE0	10
		0v			25	26	1	IN	CE1	11
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31
5	21	GPIO.21	IN	1	29	30		0v		
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26
13	23	GPIO.23	IN	0	33	34		0v		12
19	24	GPIO.24	IN	0	35	36	1	OUT	GPIO.27	27
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28
		0v			39	40	0	IN	GPIO.29	29

Abbildung 32 WiringPi Pin-Belegung

Abgesehen davon ist zu sehen, dass wiringpi eine abweichende Pin-Belegung¹⁰⁴ nutzt. Um mit wiringPi den oben beschriebenen Export zu vollziehen, kann jetzt das mitinstallierte Programm hinzugezogen werden. Dies ist insbesondere für normale Nutzer, ohne erweiterte Rechte, konstruiert¹⁰⁵. Der Export wird entsprechend durchgeführt.

```
pi@raspberrypi ~ $ gpio export 4 out
```

Für eine Stromzufuhr, muss gemäß dem Binärsystem vorangeschritten werden und am Ende des codes eine „1“ stehen. Zusätzlich muss ein **-g** zum Verweis auf die normale Pinbelegung¹⁰⁶ des Raspberry Pi auftauchen.

```
pi@raspberrypi ~ $ gpio -g write 17 1
```

6. GPIO-Ansteuerung

Das anschließende Kapitel knüpft hauptsächlich an Kapitel 5 und wird mit dem dort beschriebenen Equipment fortgeführt. Die Ansteuerung soll zum einen die Flexibilität für unterschiedliche Programmiersprachen aufzeigen und zum anderen einfache Ausgabemöglichkeiten mit LED näherbringen.

6.1. Beispielschreibprozess unter C/C++

Hierfür wird die LED-Lampe aus Kapitel 5, wieder an den Pin 7 und an Pin 14, einem Masse Pin, angeschlossen. Weiterführend ist ein wiederkehrendes Blinken der LED im Abstand von 30 Sekunden beabsichtig. Voraussetzung für diesen Ablauf ist die erfolgreiche Installation von wiringPi aus dem vorherigen Kapitel. Damit die Handhabung mit dem Code vereinfacht wird, kommt an dieser Stelle der Compiler Visual C++ ins Spiel und ist wie folgt für die Ansteuerung zu implizieren.

¹⁰⁴ Vgl. Raspberry.tips

¹⁰⁵ Vgl. Dissectionbydavid

¹⁰⁶ Vgl.Hammerschmit, Anton

```
pi@raspberrypi ~ $ vi gpio_write.cpp
```

Doch um die GPIO Bibliothek nutzen zu können, muss erst einmal die sogenannte Header-Datei¹⁰⁷ von wiringPi eingefügt werden. Jetzt ist das Nachrücken der C-Bibliothek möglich. Daraufhin, ist für die Kommunikation zwischen der Schnittstelle und dem Programm, die API¹⁰⁸ von wiringPi¹⁰⁹, miteinzubinden. Zum Zweck der Ansteuerung des gewünschten Pins muss dieser noch in den Programmcode geschrieben werden. Aus der Grafik in Kapitel 5.2, ist ersichtlich, dass Pin 7, entsprechend der wiringPi-Belegung der selben Pin-Nummer entspricht. Daher wird dieser als Ausgang deklariert. Angesichts des geplanten Blinkvorgangs, muss in den Programmcode zunächst eine Dauerschleife mit An-Funktion und Aus-Funktion eingebaut und dann eine erneute Pause von 30 Sekunden festgemacht werden. Zusammengefasst sieht der Ablauf entsprechend der Vorgaben wie unten verfasst aus.

```
#include <wiringPi.h>

#include <stdio.h>

int main() {

    if (wiringPiSetup() == -1)
        return 1;

    pinMode(7, OUTPUT);

    while(1) {

        digitalWrite(0, 1);

        delay(30);

        digitalWrite(0, 0);

        delay(30);
    }
}
```

¹⁰⁷ Vgl. **Henderson, Gordon**

¹⁰⁸ Vgl. **Henderson, Gordon**

¹⁰⁹ Vgl. **Henderson, Gordon**

Der Code muss jetzt noch in einer für den Einplatinencomputer verständlichen Sprache übersetzt werden. Daher erfolgt zunächst eine Kompilierung¹¹⁰ des Codes und dann eine Ausführung mittels Root-Berechtigung.

```
pi@raspberrypi ~ $ g++ gpio_write.cpp -o gpio_write -lwiringPi
pi@raspberrypi ~ $ sudo ./gpio_write
```

Die Schritte sind nicht vollkommen notwendig für die Funktionalität, jedoch im Hinblick auf den Lerneffekt und dem Einstiegslevel des Anwenders vorteilhaft. Deshalb sollten sie angewendet werden. Steht das nötig Equipment nicht zur Verfügung, kann der Schritt, auf die Gefahr hin die LED-Matrizen zu beschädigen, übersprungen werden.

¹¹⁰ Vgl. Ninadpchaudhari

7. Informationsausgabe mittels LED-Matrix

Da bis hierher sämtliche Vorbereitungen erfüllt sind, kann nun die geplante Ausführung auf der LED-Matrix stattfinden. Das heißt sowohl der Raspberry Pi ist voll funktionsfähig und getestet, als auch dass die LED-Matrix angeschlossen und eingeschaltet ist. Erstmals wird eine sogenannte Demo zum Anzeigen von Informationen auf der Matrix eingespielt. Im weiteren Verlauf wird der Fokus dann speziell auf die grafische und textuelle Ausgabe gerückt.

7.1. Testlauf über LED-Matrix mit C

Da die Aktualität, für diese Realisierung, stets eine tragende Rolle spielt, wird zunächst wieder das GIT aus Kapitel 5.2 aufgefrischt.

```
pi@raspberrypi $ git clone https://github.com/hzeller/rpi-rgb-led-matrix/
```

Als erstes findet ein Manövrieren zum Steuerordner für die LED-Matrix statt und ändert entsprechend die Front.

```
. pi@raspberrypi $ cd rpi-rgb-led-matrix/

. pi@raspberrypi ~/rpi-rgb-led-matrix/ $
```

Die LED-Matrix wird dann mit nachstehendem Code angesprochen. Dort erscheint eine Listenübersicht aller Anzeigeindikatoren¹¹¹ (ABB33), welche für den Einsatz mit der Matrix, verwendbar sind. Darunter gibt es noch eine entsprechende Erklärung zu den Beispielen, sogenannte Demos, die auf der LED-Matrix abgespielt werden können. Abgesehen davon ist auch ein Beispielcode hinterlegt. (ABB34)

```
. pi@raspberrypi ~/rpi-rgb-led-matrix/ $ ./led-matrix
```

¹¹¹ Vgl. MATT

```

pi@raspberrypi ~/rpi-rgb-led-matrix $ ./led-matrix
Expected required option -D <demo>
usage: ./led-matrix <options> -D <demo-nr> [optional parameter]
Options:
  -r <rows>      : Display rows. 16 for 16x32, 32 for 32x32. Default: 32
  -P <parallel>   : For Plus-models or RPi2: parallel chains. 1..3. Default: 1
  -c <chained>    : Daisy-chained boards. Default: 1.
  -L             : 'Large' display, composed out of 4 times 32x32
  -p <pwm-bits>   : Bits used for PWM. Something between 1..11
  -j            : Low jitter. Experimental. Only RPi2
  -l            : Don't do luminance correction (CIE1931)
  -D <demo-nr>   : Always needs to be set
  -d            : run as daemon. Use this when starting in
                  /etc/init.d, but also when running without
                  terminal (e.g. cron).
  -t <seconds>   : Run for these number of seconds, then exit.
                  (if neither -d nor -t are supplied, waits for <RETURN>)

```

Abbildung 33 Anzeigeoptionen 1

```

Demos, choosen with -D
  0 - some rotating square
  1 - forward scrolling an image (-m <scroll-ms>)
  2 - backward scrolling an image (-m <scroll-ms>)
  3 - test image: a square
  4 - Pulsing color
  5 - Grayscale Block
  6 - Abelian sandpile model (-m <time-step-ms>)
  7 - Conway's game of life (-m <time-step-ms>)
  8 - Langton's ant (-m <time-step-ms>)
  9 - Volume bars (-m <time-step-ms>)

Example:
  ./led-matrix -t 10 -D 1 runtext.ppm
Scrolls the runtext for 10 seconds
pi@raspberrypi ~/rpi-rgb-led-matrix $ █

```

Abbildung 34 Anzeigeoptionen 2

Wichtig für die hier aufgeführte Realisierung, ist die Größe der LED-Matrix, welche exakt 16 Pixel für die Höhe und 32 Pixel für die Breite beträgt. Zudem sind laut der Vorbereitungen drei hintereinander geschaltete LED-Matrizen aufgestellt. Dementsprechend muss der einzugebende Code dem Text unten entsprechen.

```
pi@raspberrypi ~/rpi-rgb-led-matrix/ $ sudo ./led-matrix -r 16 -c 3 -D 9
```

Ist die Eingabe korrekt eingetippt worden, erscheint nun eine Ausgabe auf den Matrizen.(ABB35). Die Demoversionen können entsprechend durchprobiert werden, um einen einwandfreien Durchlauf zu garantieren (ABB36). Damit ist der Testlauf für die LED-Matrizen abgeschlossen.

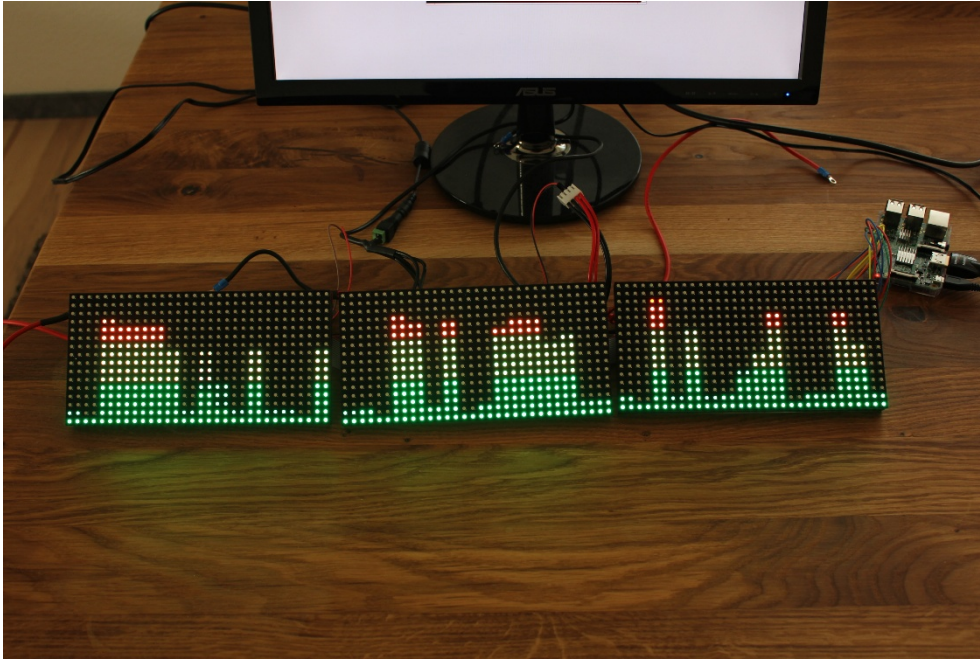


Abbildung 35 Testausgabe 1

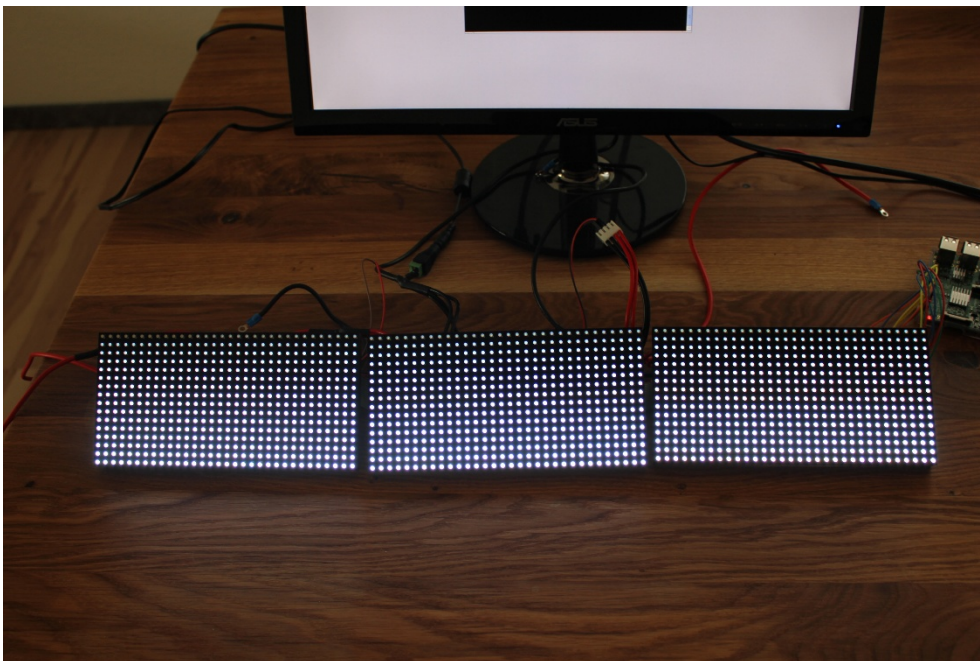


Abbildung 36 Testausgabe 2

7.2. Testlauf über LED-Matrix mit Python

Im Kapitel zuvor wird das Bildformat ppm verwendet um einen Text auf den Matrizen auszugeben. Es sollte jedoch auch möglich sein, ohne eine Bilddatei jegliche Art eines eingegebenen Textes wiederzugeben. Die einfachste Methode ist es eine Python Bibliothek einzusetzen, welche den Text in eine Bilddatei übersetzen wird und diese dann mit dem C-Code-Schema aus dem Kapitel zuvor auf den Matrizen anzeigt. Diese Bibliothek, die sogenannte Python Imaging Library¹¹² (PIL) kann über folgenden Code installiert werden.

```
pi@raspberrypi $ sudo apt-get install python-imaging
```

Jetzt erfolgt die Kontaktaufnahme zu einem Editor, der die zu übersetzende Datei erzeugt.

```
pi@raspberrypi $ nano message.py
```

Hier wird der Text wie nachstehend eingetippt. Der Text sowie die Farbe sind variabel und können entsprechend eigener Vorstellungen angepasst werden. Im Beispiel wird „I LOVE YOU“ angezeigt. Die drei Werte (255, 0, 0) bestimmen die Farbausgabe des Textes. In Anlehnung an den RGB-Farbraum, können je nach Anteil der drei Grundfarben Rot, Grün und Blau Texte eingefärbt werden¹¹³. Das heißt die Eingabe von (255, 255, 255) entspricht der Weiß. Die hier angegebenen Textteile können je nach Bedarf auch erhöht oder verringert werden. Im Beispiel unten wird das „I“ in grün, das „LOVE“ in blau und „YOU“ in rot ausgegeben. Zum Schluss wird die Datei in das Bildformat *ppm* übersetzt und als *testmitpython.ppm* gespeichert. Letzter Schritt ist der Ausgabebefehl mit dem C-Code. Hiermit ist recht einfach der Testlauf mit Python erfolgreich abgeschlossen. (ABB37)

¹¹² Vgl. Python Software Foundation

¹¹³ Vgl. Raspberry Pi Foundation

```

import os

from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw

text = (("I ", (0, 255, 0)), ("LOVE ", (0, 0, 255)), ("COKE", (255, 0, 0)))

font = ImageFont.truetype("/usr/share/fonts/truetype/freefont/FreeSans.ttf", 16)
all_text = ""
for text_color_pair in text:
    t = text_color_pair[0]
    all_text = all_text + t

print(all_text)
width, ignore = font.getsize(all_text)
print(width)

im = Image.new("RGB", (width + 30, 16), "black")
draw = ImageDraw.Draw(im)

x = 10;
for text_color_pair in text:
    t = text_color_pair[0]
    c = text_color_pair[1]
    print("t=" + t + " " + str(c) + " " + str(x))
    draw.text((x, 0), t, c, font=font)
    x = x + font.getsize(t)[0]

im.save("testmitpython.ppm")

os.system("sudo ./led-matrix testmitpython.ppm -D 1 -t 10 -c 3 -r 16")

```



Abbildung 37 Testlauf 3

7.3. Gezielte graphische Ausgabe mittels LED-Matrizen

Wie eben im Kapitel erwähnt, muss ein spezielles Dateiformat für die Ausgabe verwendet werden. Hierfür kann man sich Bildbearbeitungsprogramme zur Hilfe nehmen. Da in dieser Arbeit auch der Aspekt der Kostenersparnis eine Rolle spielt, wird das kostenlose Programm Gimp hinzugezogen. Die graphische Darstellung muss lediglich die Höhe von 16 Pixeln einhalten. Die Länge kann jedoch über die 32 Pixel hinausreichen, da der Text horizontal über die drei Matrizen scrollen wird. Ist nun eine Grafik der Wahl bereit, muss diese nur noch in das Format *.ppm* exportiert werden (ABB37).



Abbildung 38 Super Mario Grafik

Nun kann über die Konsole der folgende Befehl eingegeben werden, um das ganze auf den LED-Matrizen zu projizieren.

```
pi@raspberrypi ~/rpi-rgb-led-matrix/ $ sudo ./led-matrix -r 16 -c 3 -t 10
beispielgrafik.ppm
```

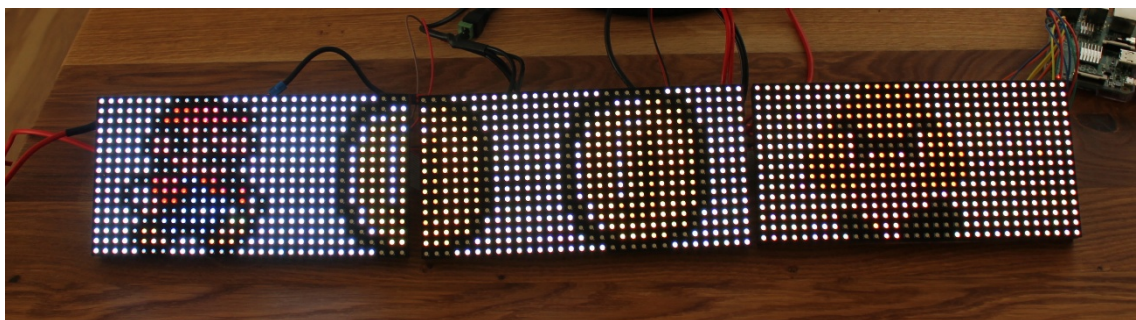


Abbildung 39 Super Mario LED

7.4. Image Viewer

Eine Möglichkeit zur graphische Ausgabe, besteht über den sogenannten *image viewer*. Auf diese Art und Weise können verschiedenste Bild-Formate gelesen und als Bild ausgegeben werden. Das Besondere hieran ist, dass auch animierte gif.-Dateien ausgegeben werden können. Für den Einsatz muss zunächst das Softwarepaket

ImageMagick installiert werden. Das Paket dient hauptsächlich zur Bearbeitung von Pixelgrafiken. Insofern ist es ein optimales Werkzeug für das Vorhaben. Dazu wird der untenstehende Code eingetippt.

```
pi@raspberrypi $ sudo aptitude install libgraphicsmagick++-dev libwebp-dev make led-image-viewer
```

Von nun können alle gängigen Bildformate auf den LED-Matrizen ausgegeben werden. Jetzt muss lediglich die entsprechende Datei in den Code integriert und wie folgt eingegeben werden. Die Anzeigooptionen aus Kapitel 7.1 gelten auch hier.

```
pi@raspberrypi $ sudo ./led-image-viewer beispiel-image-viewer.gif -r 16 -c 3 -
```

Hinter dieser Ausgabe verbergen sich insgesamt 17 Bilder, die als Verlaufsgrafik (ABB40) dargestellt sind und jeweils ein Start und Endpunkt besitzen. Der Verlauf wird so lange eingespielt, bis er mit der Return-Taste unterbrochen wird.



Abbildung 40 Super Mario GIF

Insofern ergibt sich folgende Verlaufsgrafik für die LED-Matrizen. (ABB41)

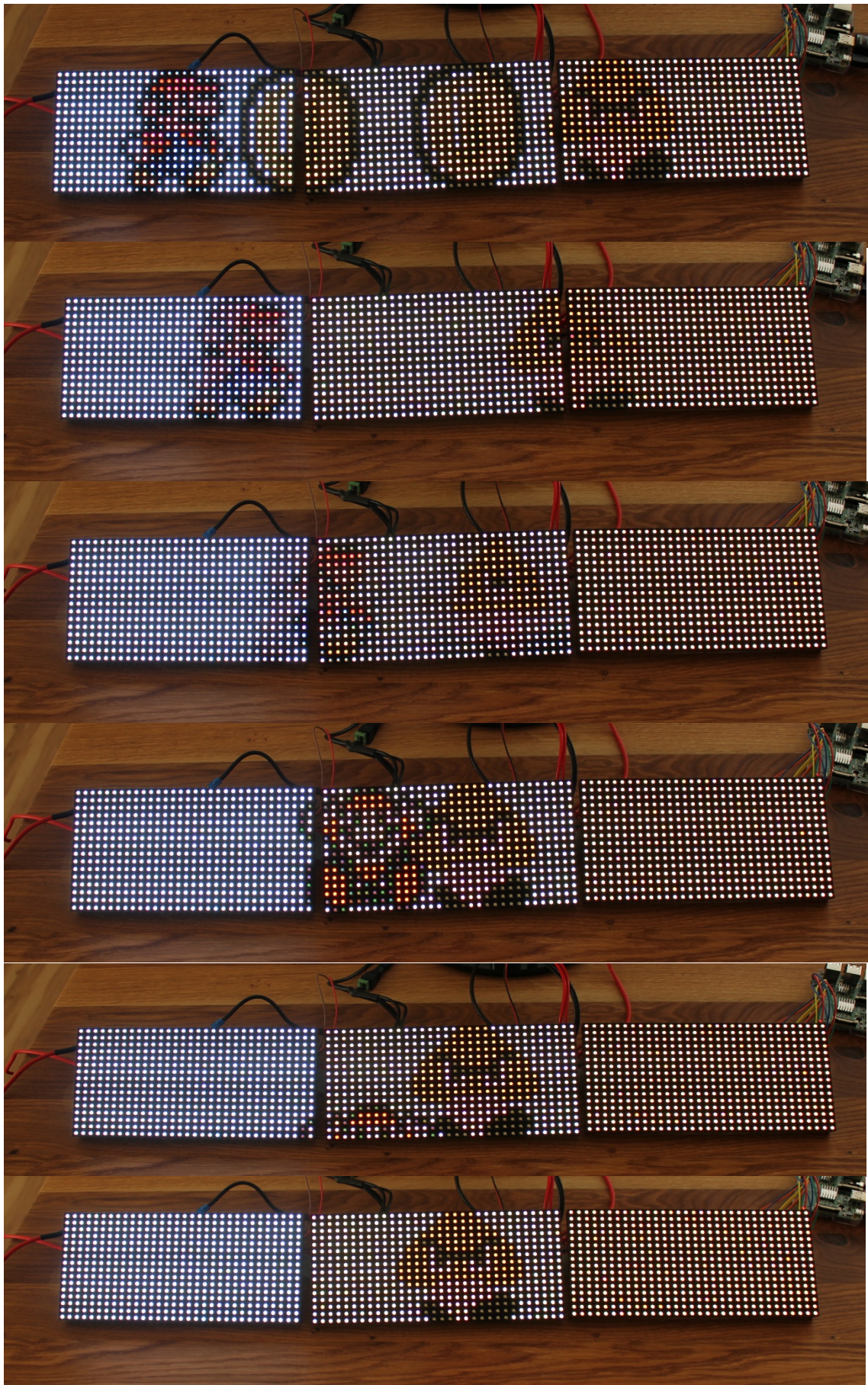


Abbildung 41 Super Mario LED

Es ist also nicht nur möglich bloße Grafiken anzuzeigen, sondern auch ein Animationsverlauf, der der Darstellung einen expliziten Gehalt verleiht. Dieser Aspekt ist unter anderem dann für Privatpersonen mit verschiedenen Etablissements interessant¹¹⁴, wenn sie ausgefallene Werbeslogans¹¹⁵ damit implizieren. Hiermit ist die graphische Ausgabe erfolgreich realisiert worden.

7.5. RSS-Feed als textuelle Ausgabe

Informationsausgaben können nicht nur als bloße Texte wiedergegeben werden, sondern beispielweise auch mithilfe des Netzwerks einen ganz neuen Gehalt erlangen. Für die angestrebte Realisierung soll ein sogenannter RSS-Feed^{116, 117} zum Einsatz kommen. Hierdurch können ausgewählte Nachrichten aus einer bestimmt definierten Quelle abgegriffen und in einer verkürzten Form auf den LED-Matrizen angezeigt werden. Die Idee für diese Umsetzung stammt von den Anzeigetafeln auf dem Times Square. Denn auch hier werden große LED-Matrizen verwendet, die sowohl textuelle als auch grafische Darstellungen präsentieren. Zur Aufzeigung der Möglichkeiten mit dem Raspberry Pi, wird im folgenden der Fokus auf die textuelle Ausgabe gelegt. Ein solcher RSS-Feed besteht aus einer XML-Datei¹¹⁸. Eine derartige Datei kann reine Inhalte strukturiert, aber ohne jegliches Layout und Navigation bereitstellen. Es gibt vielerlei Webseiten, welche die neusten Artikel über automatisch generierte Dateien zur Verfügung stellen.

7.5.1. Prototypische Umsetzung

Hierfür eignet sich unter anderem die Programmiersprache Python und ein RSS-Parser welcher dafür sorgt, dass der Inhalt eines RSS-Web-Feeds gelesen und auf dem Raspberry Pi aufbereitet werden kann. Dieser soeben angesprochene Parser, auch feedparser¹¹⁹, wird als Bibliothek in Python abgerufen. Beginnend mit dem Download

¹¹⁴ TABLICE LED, TELEBIMY

¹¹⁵ Wang, Rina

¹¹⁶ Vgl. Vidarte, Leo

¹¹⁷ Vgl. Fixtisan

¹¹⁸ Vgl. W3Schools.com

¹¹⁹ Vgl. Dragicevic, Marko

und der Installation des Paketinstallationsmanagers PIP¹²⁰ und anschließend selber Prozedur mit feedparser, kann der Vorgang im Terminal gestartet werden.

```
pi@raspberrypi $ sudo apt-get install python-pip
pi@raspberrypi $ sudo pip install feedparser
```

Ziel des Prototyps¹²¹ ist es zunächst für die Sammlung des Nachrichteninhalts zu sorgen, daher setzt sich der Code wie folgt zusammen. Der Aufbau des Prototypens besitzt in dieser Form einige Schwierigkeiten. Dementsprechend ist keine komplett erfolgreiche Ausgabe möglich, im Prinzip jedoch muss sie nach diesem Schema ablaufen.

```
#rss.py

import feedparser
feedLink = "http://www.welt.de/wirtschaft/webwelt/?service=Rss"
feed = feedparser.parse(feedLink)
items = feed["items"]
try:
    for item in items:
        print("{0}\n{1}\n".format(item["title"], item["link"]))
except ValueError:
    print(„Page not found„)
```

7.6. Auswertung der Realisierungsarbeit

Hiermit ist die Realisierung mittels Raspberry Pi abgeschlossen und sowohl eine graphische als auch eine textuelle Ausgabe erfolgreich umgesetzt worden. Dabei sind jedoch einige Problemfelder in Erscheinung getreten, welche die Realisierung teilweise

¹²⁰ Vgl. Python Software Foundation

¹²¹ Vgl. letsthinkabout.us

erschwert haben. Daher werden die Hürden in einer Abhandlungstabelle^{122, 123, 124} aufgezeigt und Möglichkeiten der Beseitigung einbezogen.

I Problemauswertung Datenträger			
Problemstellung		Lösungsansätze	Vorgehensweise
1.0	Datenträger scheint beschädigt zu sein	Überprüfen ob Betriebssystem korrekt installiert ist	Karte auf anderen Computer überprüfen auf Dateien wie start.elf und kernel.img, wenn nicht vorhanden Backup
1.1	Bootpartition enthält Dateien wie start.elf und kernel.img, startet aber nicht	Karte ist nicht kompatibel mit dem Raspberry Pi	Karte austauschen, gegebenenfalls anderen Hersteller probieren
1.2	Die Karte war bereits erfolgreich im Einsatz und funktioniert nachträglich nicht mehr	Eine unerwartet Entfernung der Karte oder ein Stromausfall unterbrechen das laufende System	Karte austauschen oder mit Backup wiederherstellen
1.3	Karte wurde während des Betriebs aus dem Lesegerät entfernt	Kartendaten können beschädigt und unbrauchbar werden	Backup oder Neuinstallation, ansonsten Karte austauschen
1.4	Lebensdauer des Datenträgers zu kurz	Auslagerung von Daten auf tiefere Ebene (Swapping) reduzieren, da genügend Speicherplatz vorhanden ist	Über die Konsoleneingabe Swapping abschalten mit <code>sudo dphys-swapfile swapoff</code> <code>sudo dphys-swapfile uninstall</code> <code>sudo update-rc.d dphys-swapfile remove</code>

¹²² Vgl. Elektronik-Kompendium.de

¹²³ Vgl. Linux Format

¹²⁴ Vgl. Orlin, Jon

Volumen der Schreibzugriffe zu hoch	Verzeichnisse die ständigen Zugriffen ausgesetzt sind, können auf den RAM ausgelagert werden	Im Verzeichnis <i>/etc/fstab</i> folgende Einträge hinzufügen <i>none/var/run tmpfs size=5M,noatime 00</i> <i>none/var/run tmpfs size=5M,noatime 00</i>
Vermeidungsstrategie - Raspberry Pi über die graphische Oberfläche herunterfahren oder bei SSH-Verbindung den Konsolenbefehl: <i>sudo shutdown -h now</i> - Beim Start auch die LEDs des Raspberry beobachten, da diese in den meisten Fällen Hinweise zum Problem geben - Daten und Schreibzugriffe die nicht gebraucht werden, sollten beseitigt werden		

Tabelle 4 Problemauswertung Datenträger

II Problemauswertung Display			
Problemstellung		Lösungsansätze	Vorgehensweise
2.0	Das an den Raspberry Pi angeschlossene Display bleibt beim booten aus	Displayeinstellungen sind zum Boot nicht optimal eingerichtet	Raspberry ausschalten und erneut starten, währenddessen auf die LED des Raspberry achten
2.1	Display bleibt nach dem zweiten Versuch aus und die LED liefert keine Hinweise	Kommunikationsfehler zwischen Display und dem Raspberry Pi fehlerhaft daher Änderung in der Konfiguration nötig	Karte auf anderem Computer nach Datei <i>config.txt</i> untersuchen, Datei mit Editor öffnen und dem Text <i>hdmi_safe=1</i> hinzufügen
2.2	Auf dem Display sind Balken, Pixel oder Schleier zu sehen	Kabel beschädigt oder Signalweg zu lang	Kabel austauschen, kürzeres Kabel oder wie in 2.1 <i>config.txt</i> öffnen und Text <i>config_hdmi_boost=4</i> einfügen
2.3	Raspberry Desktop wird nur teilweise oder unscharf abgebildet	Der Wert für den Overscan ist nicht korrekt und muss	Werte in der Datei aus 2.1 <i>config.txt</i> ändern auf

		geändert werden bis das Bild vollkommen zu sehen ist	Overscan_left=10 overscan_right=10 overscan_top=30 overscan_bottom=30
2.4	HDMI-Auflösung fehlerhaft	Displayeinstellungen sind nicht optimal eingerichtet	Display zuerst einschalten und dann Raspberry Pi dann starten
Vermeidungsstrategie: • Kabelverbindung zwischen Display und Einplatinencomputer genauestens überprüfen • Raspberry erst an Stromversorgung schließen wenn die Verbindung zwischen den Komponenten besteht • Display-Einstellungen in der config.txt Datei überprüfen			

Tabelle 5 Problemauswertung Display

III Problemauswertung Stromversorgung			
Problemstellung		Lösungsansätze	Vorgehensweise
3.0	Die rote LED des Raspberry Pi blinkt unregelmäßig	Es besteht ein Problem mit der Stromversorgung	Stromkabel auf Brüche untersuchen und sichergehen, dass es stabile 5V liefert
3.1	Stromversorgung ist vorher erfolgreich geflossen, scheint aber im späteren Verlauf nicht mehr vorhanden zu sein	Patine des Raspberry mit einem Voltmeter untersuchen	In der rechten unteren Ecke der Rückseite befinden sich die Messpunkte, die Spannung sollte nicht unter 4,97 V liegen
Vermeidungsstrategie: • Stromversorgung immer erst anschließen, nachdem alle Verbindungen zwischen Komponenten auf die Richtigkeit überprüft wurden • Raspberry Pi nicht von der Stromversorgung trennen, wenn es noch im Betrieb ist			

Tabelle 6 Problemauswertung Stromversorgung

IV Problemauswertung LED-Matrix			
Problemstellung		Lösungsansätze	Vorgehensweise
4.0	LED-Matrix geht nicht an	Verbindungen überprüfen	Beim Anschalten des Raspberry Pi auf kurze Leuchtausgabe achten, sonst Stromkabel austauschen
	LED-Matrix zeigt nur teilweise die gewünschte Ausgabe an	Codeeingabe überprüfen	Besonders die Höhe darf nicht 16 Pixel übersteigen, daher ist der Code nochmal durchsehen
Vermeidungsstrategie: • Stromkabel und Jumperkabel immer auf feste und korrekte Verbindung überprüfen • Kabel austauschen wenn sie zu sehr abgenutzt sind			

Tabelle 7 Problemauswertung LED-Matrix

8. Mögliche Realisierungsansätze im öffentlichen und privaten Bereich

Der nächste Abschnitt behandelt Ideen für Realisierungsarbeiten mit dem Raspberry Pi und dient lediglich als Anstoß für Konzepte, die damit theoretisch funktionieren könnten. Genau solche Fälle sollen kurz auf die Machbarkeit untersucht werden.

8.1. Öffentlichen Bereich – Trafficmanagement und Parksystem

Der erste Entwurf ist im Straßenverkehr ausgelegt und konzentriert sich insbesondere auf die Darstellung mit LED-Matrizen. Gerade Straßenverkehrszeichen auf Autobahnen, erfordern oftmals wechselfähige graphische Darstellungsmöglichkeiten¹²⁵, weil unerwartet Verkehrssituationen entstehen (ABB42). Das heißt beispielsweise ereignet sich irgendwo auf der Strecke ein Verkehrsunfall. Jetzt sind entsprechende Regulationsstellen für den Verkehr in der Lage, weitaus vor der betroffenen Strecke, die Geschwindigkeit nachfolgender Fahrer zu reduzieren. Dadurch kann Stau gedämmt werden. Die vielseitigen Eigenschaften des Einplatinencomputers Raspberry Pi, könnten an dieser Stelle äußerst hilfreich sein. Da dieser unter anderem einen Dauerbetrieb¹²⁶ ermöglicht und trotzdem vergleichsweise geringe Strommengen verbraucht, wäre der kleine Einplatinencomputer halbwegs einsetzbar. Lediglich eine sichere und wetterfeste Position müsste errichtet werden. Nützlich an dieser Stelle, ist der Ethernet-Anschluss, da hierdurch die Informationen in Echtzeit und von jeder Position weltweit veränderbar sind. Weiterhin wäre die Nutzung der GPIO-Schnittstelle in Kombination mit Sensoren eine Option um Fahrzeugpositionen teilweise einzuordnen. Dies wäre auch hinsichtlich eines Parksystems¹²⁷ interessant (ABB43). Denn durch Sensoren könnten noch freie Parkplätze erfasst, an den Computer, in diesem Fall an den Raspberry Pi gesendet werden. Nun könnten diese Informationen mit einem Code überprüft und entsprechende Pfeile und Nummern mittels LED-Matrizen als Wegweiser für Fahrzeuge zum Einsatz kommen. Das problematische an beiden Ansätzen ist die Gewährleistung einer stabilen Stromzufuhr¹²⁸ für den Raspberry. Denn wird sie unterbrochen, gerät der gesamte Prozess ins Schwanken. Zudem wäre der Umfang der

¹²⁵ Vgl. Bremicker Verkehrstechnik GmbH & Co. KG

¹²⁶ Vgl. Manuel

¹²⁷ Vgl. ParkSol

¹²⁸ Vgl. Elektronik-Kompodium.de

Informationen¹²⁹ mit der Leser-Schreibrate des Raspberry nicht realisierbar. Das ergibt sich zum einen daraus, dass der Raspberry Pi eine SD-Karte bzw. ein Micro-SD-Karte verwendet und zum anderen aus der Größe des Arbeitsspeichers, welcher Prozesse eines solchen Umfangs nicht verarbeiten können würde. Daher ist aus jetziger Sicht die Machbarkeit mit dem Einplatinencomputer nicht möglich.



Abbildung 42 Verkehrsmanagement



Abbildung 43 Parksystemmanagement

8.2. Privater Bereich – Reklame und Innendesign

Der zweite Entwurf konzentriert sich auf die Realisierung im privaten Bereich. Die vorherigen Kapitel haben gezeigt, dass der Raspberry eine kostengünstige Option für allerlei Projekte darstellt. Zudem können verschiedenen Arten von graphischen und textuellen Ausgaben erzeugt werden. Daher eignet sich eine Umsetzung mittels Raspberry Pi und LED-Matrizen hervorragend für kleine Etablissements, die damit an unübersichtlichen Straßen auf sich aufmerksam machen wollen^{130, 131}. Besitzer solcher Etablissements können selbst Eyecatcher (ABB44) entwickeln, ohne ein breitgefächertes Vorwissen aufzuweisen. Natürlich kann der Aufbau auch für das Innenleben bzw. Innendesign in einem Raum¹³² verwendet werden. Unter anderem könnten die LED-Matrizen für einen, vorher zeitlich gesetzten Farbwechsel sorgen und somit eine bestimmte Atmosphäre schaffen (ABB45). Durch solche Akzente erhält das Etablissement einen hohen Wiedererkennungswert bei Kunden. Andererseits könnten

¹²⁹ Vgl. Elektronik-Kompendium.de

¹³⁰ Vgl. Braford, Shanti

¹³¹ Vgl. TABLICE LED, TELEBIMY

¹³² Vgl. INARAY Design Group

die LED-Matrizen auch hier genutzt werden, um beispielsweise textuell zu bestimmten Uhrzeiten Preisinformationen¹³³ zu schalten, welche die Aufmerksamkeit der Kunden auf sich zieht. Nun wie im Kapitel vorher schon angesprochen ist auch hier das Problem der stabilen Stromzufuhr. Jedoch müsste die Hardware nur eine bestimmte Anzahl an Stunden pro Tag funktionieren. Das heißt gerade tagsüber kann auf solche Hilfsmittel verzichtet werden. Für die Lese und Schreibrate ist entsprechend mit keiner Problematik zu rechnen, da diese kleinen Zugriffe keine hohen Rechenleistungen erfordern. Deswegen ist der Raspberry sowohl in der Anschaffung als auch in der Handhabung ideal für solche Einsätze.



Abbildung 44
Reklametafel



Abbildung 45 Wechselbeleuchtung LED

¹³³ Vgl. praeniteo GmbH

9. Resümee und Ausblick

Die Realisierung mittels Raspberry Pi und LED-Matrix ist damit abgeschlossen. Rückblickend ist zu sagen, dass die Umsetzung auf vielen verschiedenen Wegen lösbar wäre. Im Rahmen der angestrebten Kriterien in dieser Arbeit ist der beschriebene Ablauf besonders für Studenten und Privatpersonen geeignet. Im Bezug zum Informatik Studium ist zu sagen, dass gerade der Einsatz vom Raspberry Pi einen unheimlich großen Lerneffekt sowohl im Hardware- als auch im Softwarebereich haben kann. In dieser Arbeit wurde lediglich die Ansteuerung zur LED-Matrix erprobt, aber klar ist, dass es schier unendlich viele Möglichkeiten gibt. Der Einplatinencomputer ist in der Handhabung einfach und koppelt gleichzeitig eine gewisse Komplexität in jedes Projekt. Das erlaubt ein unbeschwertes Experimentieren, ohne absolut vertiefte Kenntnisse zu haben. Gleichzeitig ist für Experten genug Spielraum nach oben hin.

Abgesehen davon sollte die Arbeit als Versuchsprotokoll der eigenen Durchführungsprozedur dienen. Der Aufbau der Elemente ist nun in stark vereinfachter Form fähig fließende Texte und Grafiken darzustellen, wie sie auf dem Times Square zu sehen sind. Hinzu kommen die moderaten Anschaffungskosten für die gesamte Hardware. Neben der kompakten Form und den vielseitigen Anschlussmöglichkeiten, verbraucht der Einplatinencomputer nur sehr wenig Strom.

Explizit die GPIO-Pins machen den Raspberry so besonders und haben unter anderem auch diese Realisierung ermöglicht. In Kapitel 7.5 wird jedoch auch deutlich, wie Kleinigkeiten zu Startschwierigkeiten führen können. Gerade Anfänger könnten hier scheitern oder ohne Hilfe durch Experten und der Community scheitern.

An Vielseitigkeit für Programmiersprachen fehlt es dem Raspberry nicht. Kapitel 5-7 führen nur einige der verwendbaren Sprachen auf, im Hinblick auf die eigenen Kenntnisse und der eines Einsteigers aus jeglichem Umfeld, ist gerade die Wahl mit dem C-Code am einfachsten. Das heißt aber nicht, dass die Verwendung anderer Programmiersprachen ein Problem darstellt, sondern an diesen Stellen zu umständlich gewesen wären.

Angesichts der Experimentierfreudigkeit, die durch den Raspberry PI geweckt wird, ist die Überlegung von Projekten im privaten und im öffentlichen Bereich interessant. Nachweislich realisierbar ist mit dieser Arbeit jedoch nur die Privatsektion, wie aus eigenen Kenntnissen in Kapitel 8 durchgeblendet wurde.

Im Hinblick auf die verwendete Hardware und Software, ist davon auszugehen, dass diese auch in Zukunft leicht und preisgünstig erwerbbar sein werden. Überdies hinaus könnte die Raspberry Pi Foundation neue verbesserte Einplatinencomputer entwickeln, welche unter Umständen noch einfacher zu bedienen werden und Verbrauchern einen noch bequemeren Einstieg gewähren. Auch ist es ihnen möglich Projekte zu entwickeln, die auf dem Raspberry Pi basieren und diese dann zu verkaufen. Infolgedessen kann dieser Umstand als Motivationshintergrund für den Aufbau entsprechender Projekte genutzt werden.

Aus eigener Sicht hat die Realisierungsarbeit und Auseinandersetzung mit den vielen kleinen Aspekten, die den Raspberry Pi ausmachen, einen hohen Integrationsgrad für die eigene Zukunft erbeutet. Darüber hinaus hat das Studium der Informatik die Wissensgrundlage für die Bearbeitung dieser Arbeit geliefert und dadurch viele Gesichtspunkte erleichtert. Und gleichzeitig hat der Prozess in sich selbst bestehendes Wissen erweitert.

Zusammenfassend lässt sich sagen, dass die Projektrealisierung ein vollständiges und erfolgreiches Resultat hervorgebracht hat. Aus vielen verschiedenen Einzelteilen ist ein vollwertiges System entstanden, dass sowohl für Lehrinstitutionen, als auch Selbstständige mit entsprechendem Etablissement verwendet werden könnte. Wohingegen der öffentliche Sektor zum jetzigen Zeitpunkt problematisch für den Einplatinencomputer ist. Da es hier sicherlich weitaus leistungsstärkere Systeme gibt die eine Informationsausgabe ermöglichen.

Literaturverzeichnis

Hinweis

Die Quellen bezüglich dieses Themengebietes sind größtenteils aus dem Internet entnommen, da die Technologie noch nicht ausgereift ist. Angesichts dessen können die Angaben zu den Autoren unpräzise sein und werden dann mit den entsprechenden Unternehmen oder Entwicklergruppen angegeben.

- [1] **Gartner, Inc.** [Online abgerufen am 23.01.2016] URL:
<http://www.gartner.com/smarterwithgartner/whats-new-in-gartners-hype-cycle-for-emerging-technologies-2015/>
- [2] **Marshall Gary** [Online abgerufen am 23.01.2016] URL:
<http://www.techradar.com/news/computing/just-how-popular-can-the-raspberry-pi-get—1215103>
- [3] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/about/>
- [4] **Kling Bernd** [Online abgerufen am 23.01.2016] URL:
<http://www.zdnet.de/88142428/die-geschichte-des-raspberry-pi/?PageSpeed=noscript>
- [5] **Marko Dragicevic** [Online abgerufen am 23.01.2016] URL:
<http://www.raspberry-pi-geek.de/Magazin/2014/01/Spielkonsolen-Klassiker-auf-dem-Raspberry-Pi-emulieren>
- [6] **Kim Rixecker** [Online abgerufen am 23.01.2016] URL:
<http://t3n.de/news/raspberry-pi-projekte-434138/>
- [7] **George Michael** [Online abgerufen am 23.01.2016] URL:
<http://www.computer-history.info/Page4.dir/pages/PDP.1.dir/>
- [8] **Andrea, Hackbarth** [Online abgerufen am 23.01.2016] URL:
<http://www.embedded-design.net/allgemein/marktuebersichtsystem-on-chip-soc/>
- [9] **Mattern, Friedemann (Hrsg.); Springer Verlag (20.Juni 2007), Die Informatisierung des Alltags: Leben in smarten Umgebungen**

- [10] **B.A.S. Verkehrstechnik AG** [Online abgerufen am 23.01.2016] URL:
<http://www.bas-verkehr.de/>
- [11] **Computronics Ltd 2009** [Online abgerufen am 23.01.2016] URL:
<http://www.computronics.biz/industry/airport.html>
- [12] **itdz-berlin** [Online abgerufen am 23.01.2016] URL:
https://www.itdz-berlin.de/dokumente/itdz_berlin_whitepaper_green_it.pdf
- [13] **Raspishop.de** [Online abgerufen am 23.01.2016] URL:
<https://www.raspishop.de/Raspberry-Pi-Model-B-original-UK>
- [14] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/community/>
- [15] **TABLICE LED, TELEBIMY** [Online abgerufen am 23.01.2016] URL:
<https://www.youtube.com/watch?v=pdMxixVekmc>
- [16] **Elektronik-Kompendium.de** [Online abgerufen am 23.01.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/1910071.htm>
- [17] **TABLICE LED, TELEBIMY** [Online abgerufen am 23.01.2016] URL:
<https://www.youtube.com/watch?v=AYhyKv6cP5Q>
- [18] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/about/>
- [19] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
ebd.
- [20] **Arthur, Charles** [Online abgerufen am 23.01.2016] URL:
<http://www.theguardian.com/technology/2013/aug/30/pc-boom-over-tablets-smartphones>
- [21] **Damani Viranch** [Online abgerufen am 23.01.2016] URL:
<http://techpp.com/2015/11/21/state-of-smartphone-soc-market-explained/>
- [22] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/about/>
- [23] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/blog/the-alpha-boards-are-here/>
- [24] **Upton, Liz** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/blog/two-million/>
- [25] **Upton, Liz** [Online abgerufen am 23.01.2016] URL:
ebd.
- [26] **Kofler, Michael**; Kühnächst Charly; Scherbeck Christoph; Rheinwerk Computing;
Auflage: 2 (31. August 2015) Raspberry Pi: Das umfassende Handbuch. Komplett in
Farbe – inkl. Schnittstelle, Schaltungsaufbau, Steuerung mit Python

- [27] **Wood, Roy** [Online abgerufen am 23.01.2016] URL:
<http://www.wired.com/2012/03/raspberry-pi-more-popular-than-lady-gaga/>
- [28] **Kofler, Michael** [Online abgerufen am 23.01.2016] URL:
<http://pi-buch.info/?p=25>
- [29] **Devicespecifications.com** [Online abgerufen am 23.01.2016] URL:
<http://www.devicespecifications.com/de/model-cpu/95ba272a>
- [30] **Anthony, Sebastian** [Online abgerufen am 23.01.2016] URL:
<http://www.extremetech.com/computing/126235-soc-vs-cpu-the-battle-for-the-future-of-computing>
- [31] **Raspberry.tips** [Online abgerufen am 24.01.2016] URL:
<https://raspberry.tips/raspberrypi-tutorials/raspberry-pi-raspbian-auf-windows-emulieren-qemu/>
- [32] **Adafruit Industries** [Online abgerufen am 23.01.2016] URL:
<https://www.adafruit.com/datasheets/pi-specs.pdf>
- [33] **Elektronik-Kompendium.de** [Online abgerufen am 23.01.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/1905251.htm>
- [34] **Hammerschmit, Anton** [Online abgerufen am 23.01.2016] URL:
<http://rasberrypiguide.de/>
- [35] **Ziegelwanger, Werner** [Online abgerufen am 22.01.2016] URL:
<https://developer-blog.net/hardware/raspberrypi/raspberry-pi-b-test-stromverbrauch/>
- [36] **Dragicevic, Marko** [Online abgerufen am 22.01.2016] URL:
<http://www.raspberry-pi-geek.de/Magazin/2014/01/Die-richtige-Programmiersprache-fuer-den-RasPi-finden>
- [37] **Tutorials-RaspberryPi** [Online abgerufen am 22.01.2016] URL:
<http://tutorials-raspberrypi.de/raspberry-pi-sensoren-uebersicht-die-40-wichtigsten-module/>
- [38] **Tutorials-RaspberryPi.** [Online abgerufen am 22.01.2016] URL:
ebd.
- [39] **MATT** [Online abgerufen am 22.01.2016] URL:
<http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>

- [40] **MATT** [Online abgerufen am 22.01.2016] URL:
ebd.
- [41] **Elektronik-Kompendium.de** [Online abgerufen am 22.01.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/2006031.htm>
- [42] **Christian Klaub** [Online abgerufen am 22.01.2016] URL:
<http://www.golem.de/news/picade-mini-raspberry-pi-als-minispielautomat-1211-95511.html>
- [43] **Schwarz, Andreas** [Online abgerufen am 23.01.2016] URL:
<http://www.mikrocontroller.net/articles/LED-Matrix>
- [44] **Ryniker, Richard** [Online abgerufen am 22.01.2016] URL:
<http://www.raspberry-pi-geek.de/Magazin/2013/05/Tricks-zum-Programmieren-der-GPIO-Schnittstelle>
- [45] **Ryniker, Richard** [Online abgerufen am 22.01.2016] URL:
ebd.
- [46] **Ryniker, Richard** [Online abgerufen am 22.01.2016] URL:
ebd.
- [47] **Schwarz, Andreas** [Online abgerufen am 23.01.2016] URL:
<http://www.mikrocontroller.net/articles/LED-Matrix>
- [48] **Schwarz, Andreas** [Online abgerufen am 23.01.2016] URL:
ebd.
- [49] **Richter, Peter** [Online abgerufen am 23.01.2016] URL:
<http://www.zeit.de/1968/16/strom-der-ewig-kreist/seite-2>
- [50] **Dransfeld, Damon** [Online abgerufen am 23.01.2016] URL:
<http://www.tacticalcode.de/2013/01/led-matrix-grundlagen-mit-multiplexing.html>
- [51] **TenDance** [Online abgerufen am 23.01.2016] URL:
<http://www.computerbase.de/forum/showthread.php?t=1046013>
- [52] **TenDance** [Online abgerufen am 23.01.2016] URL:
ebd.
- [53] **Dragicevic, Marko** [Online abgerufen am 27.01.2016] URL:
<http://www.raspberry-pi-geek.de/Magazin/2013/05/Distributionen-fuer-den-Raspberry-Pi-im-Ueberblick>
- [54] **Britz, Christian; Golde, Nico; u.a.** [Online abgerufen am 23.01.2016] URL:
https://www.debian.org/doc/manuals/debian-faq/ch-basic_defs.de.html

- [55] **Raspberry Pi Foundation** [Online abgerufen am 25.01.2016] URL:
<https://www.raspbian.org/>
- [56] **Biglesp** [Online abgerufen am 21.02.2016] URL:
<https://www.element14.com/community/polls/2103>
- [57] **Steuer, Richard** u Kohne Joerg [Online abgerufen am 22.02.2016] URL:
<http://www.gnu.org/gnu/linux-and-gnu.de.html>
- [58] **Thompson, John** [Online abgerufen am 23.01.2016] URL:
<http://www.cnet.com/how-to/how-to-install-android-2-3-on-the-raspberry-pi/>
- [59] **Westaway, Luke** [Online abgerufen am 23.01.2016] URL:
<http://www.cnet.com/news/raspberry-pi-2-model-b-is-a-quad-core-upgrade-on-sale-today/>
- [60] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/blog/price-cut-raspberry-pi-model-b-now-only-25/>
- [61] **Thoma, Jörg** [Online abgerufen am 23.01.2016] URL:
<http://www.golem.de/news/raspberry-b-im-test-sparsamer-nachfolger-fuer-mehr-bastelspass-1408-108504-2.html>
- [62] **T. Nick** [Online abgerufen am 23.01.2016] URL:
http://www.phonearena.com/news/PhoneArena-Awards-2011-Best-smartphone_id24725
- [63] **Thoma, Jörg** [Online abgerufen am 23.01.2016] URL:
<http://www.golem.de/news/miniplatine-ist-das-raspberry-pi-wirklich-offene-hardware-1403-104564.html>
- [64] **Raspberry Pi Foundation** [Online abgerufen am 02.03.2016] URL:
<https://www.raspberrypi.org/blog/starting-a-business-with-a-raspberry-pi/>
- [65] **Elektronik-Kompendium.de** [Online abgerufen am 23.01.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/1905261.htm>
- [66] **SASCHA** [Online abgerufen am 02.03.2016] URL:
<http://saschas-blog.net/feed/2013/05/raspberry-pi-fast-unendliche-moglichkeiten/>
- [67] **Henderson, Gordon** [Online abgerufen am 02.03.2016] URL:
<https://projects.drogon.net/a-quick-look-at-the-new-raspberry-pi-model-b/>
- [68] **SASCHA** [Online abgerufen am 02.03.2016] URL:
<http://saschas-blog.net/feed/2013/05/raspberry-pi-fast-unendliche-moglichkeiten/>

- [69] **Raspberry Pi Foundation** [Online abgerufen am 19.02.2016] URL:
https://www.raspbian.org/RaspbianFAQ#Raspbian_FAQ
- [70] **Raspberry Pi Foundation** [Online abgerufen am 19.02.2016] URL:
<https://www.raspberrypi.org/downloads/raspbian/>
- [71] **Sharpened Productions** [Online abgerufen am 19.02.2016] URL:
<http://techterms.com/definition/torrent>
- [72] **Raspberry Pi Foundation** [Online abgerufen am 19.02.2016] URL:
<https://www.raspberrypi.org/downloads/raspbian/>
- [73] **Hoffman, Chris** [Online abgerufen am 23.01.2016] URL:
<http://www.howtogeek.com/235596/whats-the-difference-between-fat32-exfat-and-ntfs/?PageSpeed=noscript>
- [74] **Schanze, Robert** [Online abgerufen am 19.02.2016] URL:
<http://www.com-magazin.de/tipps-tricks/software/sd-sdhc-sdxc-karten-formatieren-882284.html>
- [75] **SD Association** [Online abgerufen am 19.02.2016] URL:
https://www.sdcard.org/downloads/formatter_4/
- [76] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/documentation/linux/filesystem/backup.md>
- [77] **ITWissen.info** [Online abgerufen am 23.01.2016] URL:
<http://www.itwissen.info/definition/lexikon/secure-shell-SSH-SSH-Protokoll.html>
- [78] **Tatham, Simon** [Online abgerufen am 23.01.2016] URL:
<http://www.putty.org/>
- [79] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/documentation/linux/usage/users.md>
- [80] **Raspberry Pi Foundation** [Online abgerufen am 19.02.2016] URL:
<https://www.raspberrypi.org/documentation/configuration/raspi-config.md>
- [81] **Munteanu, Llie-Alexandru** [Online abgerufen am 19.02.2016] URL:
<http://www.digitalcitizen.life/simple-questions-what-bios-what-does-it-do>
- [82] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/blog/add-your-raspberry-pi-to-the-rastrack-map/>
- [83] **Raspberry Pi Foundation** [Online abgerufen am 19.02.2016] URL:
<https://www.raspberrypi.org/documentation/usage/terminal/>
- [84] **Kampert, Daniel**; Rheinwerk Computing; Auflage: 3 (8. Oktober 2015); Raspberry Pi: Der praktische Einstieg zum Raspberry Pi 2 und allen Vorversionen
- [85] Ebd.

- [86] **Tatham, Simon** [Online abgerufen am 05.03.2016] URL:
<http://www.putty.org/>
- [87] **Assigned Numbers Authority** [Online abgerufen am 05.03.2016] URL:
<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>
- [88] **MaxEmbedded** [Online abgerufen am 23.01.2016] URL:
<http://maxembedded.com/2015/04/raspberry-pi-headless-setup/>
- [89] **GitHub, Inc.** [Online abgerufen am 05.03.2016] URL:
<https://github.com/adafruit/Adafruit-Pi-Finder>
- [90] **Savie, Ravi** [Online abgerufen 05.03.2016] URL:
<http://www.tecmint.com/useful-basic-commands-of-apt-get-and-apt-cache-for-package-management/>
- [91] **Raspberry Pi Foundation** [Online abgerufen am 23.01.2016] URL:
<https://www.raspberrypi.org/documentation/usage/terminal/>
- [92] **Savie, Ravi** [Online abgerufen am 23.01.2016] URL:
<http://www.tecmint.com/useful-basic-commands-of-apt-get-and-apt-cache-for-package-management/>
- [93] **Henderson, Gordon** [Online abgerufen am 07.03.2016] URL:
<https://projects.drogon.net/raspberry-pi/wiringpi/functions/>
- [94] **Henderson, Gordon** [Online abgerufen am 07.03.2016] URL:
ebd.
- [95] **Hammerschmit, Anton** [Online abgerufen am 03.03.2016] URL:
<http://raspberrypiguide.de/howtos/raspberry-pi-gpio-how-to/>
- [96] **Hammerschmit, Anton** [Online abgerufen am 03.03.2016] URL:
ebd.
- [97] **Kampert, Daniel**; Rheinwerk Computing; Auflage: 3 (8. Oktober 2015); Raspberry Pi: Der praktische Einstieg zum Raspberry Pi 2 und allen Vorversionen
- [98] **Torvalds, Charles** [Online abgerufen am 01.03.2016] URL:
<http://www.askapache.com/security/chmod-stat.html>
- [99] **Kampert, Daniel**; Rheinwerk Computing; Auflage: 3 (8. Oktober 2015); Raspberry Pi: Der praktische Einstieg zum Raspberry Pi 2 und allen Vorversionen
- [100] **Elektronik-Kompendium.de** [Online abgerufen am 23.01.2016] URL:
<http://www.elektronik-kompendium.de/sites/dig/0208051.htm>
- [101] **Henderson, Gordon** [Online abgerufen am 01.03.2016] URL:
<https://projects.drogon.net/raspberry-pi/wiringpi/functions/>

- [102] **Henderson, Gordon** [Online abgerufen am 23.02.2016] URL:
<https://projects.drogon.net/raspberry-pi/wiringpi/functions/>
- [103] **Torvalds, Linus; , Pearce, Shawn O.; Hamano, Junio C.; u.a.**
[Online abgerufen am 23.01.2016] URL:
<https://git-scm.com/about>
- [104] **Raspberry.tips** [Online abgerufen am 23.03.2016] URL:
<https://raspberrypi-news.belegung-gpio-fuer-raspberry-pi-b-und-wiringpi/>
- [105] **Dissectionbydavid** [Online abgerufen am 01.03.2016] URL:
<https://dissectionbydavid.wordpress.com/2013/10/21/raspberry-pi-using-gpio-wiringpi-without-root-sudo-access/>
- [106] **Hammerschmit, Anton** [Online abgerufen am 01.03.2016] URL:
<http://raspberrypiguide.de/howtos/raspberry-pi-gpio-how-to/>
- [107] **Henderson, Gordon** [Online abgerufen am 01.03.2016] URL:
<https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/>
- [108] **Henderson, Gordon** [Online abgerufen am 05.03.2016] URL:
<https://projects.drogon.net/raspberry-pi/wiringpi/functions/>
- [109] **Henderson, Gordon** [Online abgerufen am 01.03.2016] URL:
<https://github.com/WiringPi/WiringPi/blob/master/wiringPi/wiringPi.h>
- [110] **Ninadpchaudhari** [Online abgerufen am 23.02.2016] URL:
<https://www.raspberrypi.org/forums/viewtopic.php?t=41973&p=339458>
- [111] **MATT** [Online abgerufen am 08.02.2016] URL:
<http://www.raspberrypi-spy.co.uk/2014/09/32x32-led-matrix-on-the-raspberry-pi/>
- [112] **Python Software Foundation** [Online abgerufen am 09.03.2016] URL:
<https://pypi.python.org/pypi/PIL>
- [113] **Raspberry Pi Foundation** [Online abgerufen am 15.03.2016] URL:
<https://www.raspberrypi.org/learning/getting-started-with-the-sense-hat/worksheet/>
- [114] **TABLICE LED, TELEBIMY** [Online abgerufen am 15.03.2016] URL:
<https://www.youtube.com/watch?v=U7CC-TbtVoA>
- [115] **Wang, Rina** [Online abgerufen am 13.03.2016] URL:
<https://www.youtube.com/watch?v=XaO3LgeGo8s>
- [116] **Vidarte, Leo** [Online abgerufen am 16.03.2016] URL:
<https://www.youtube.com/watch?v=Xjc1pk05hlw>
- [117] **Fixtisan** [Online abgerufen am 16.03.2016] URL:
<https://www.youtube.com/watch?v=Zv0RYwwjQdQ>

- [118] **W3Schools.com** [Online abgerufen am 23.01.2016] URL:
http://www.w3schools.com/xml/xml_rss.asp
- [119] **Dragicevic, Marko** [Online abgerufen am 17.02.2016] URL:
<http://www.raspberry-pi-geek.de/Magazin/2013/06/Webbasierte-Feedreader-auf-dem-RasPi-hosten/%28offset%29/2>
- [120] **Python Software Foundation** [Online abgerufen am 17.02.2016] URL:
<https://pypi.python.org/pypi/pip>
- [121] **Elektronik-Kompendium.de** [Online abgerufen am 17.02.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/2101221.htm>
- [122] **Linux Format** [Online abgerufen am 17.02.2016] URL:
<http://www.techradar.com/how-to/computing/how-to-fix-raspberry-pi-boot-problems-1310697>
- [123] **Orlin, Jon** [Online abgerufen am 05.02.2016] URL:
<http://techcrunch.com/2012/10/21/getting-started-with-the-raspberry-pi-is-not-as-easy-as-pie/>
- [124] **Bremicker Verkehrstechnik GmbH & Co. KG** [Online abgerufen am 19.02.2016] URL:
http://www.bremicker-vt.de/sites/default/files/styles/fullscreen/public/bilder/led_schilderkombo.jpg?itok=ElhATeUH
- [125] **Manuel** [Online abgerufen am 09.02.2016] URL:
<https://www.datenreise.de/raspberry-pi-stabiler-24-7-dauerbetrieb/>
- [126] **ParkSol** [Online abgerufen am 09.02.2016] URL:
<http://www.parksol-parking.com/en/products/led-panels/1/led-display-11>
- [127] **Elektronik-Kompendium.de** [Online abgerufen am 09.02.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/1912111.htm>
- [128] **Elektronik-Kompendium.de** [Online abgerufen am 14.02.2016] URL:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/2002011.htm>
- [129] **Braford, Shanti** [Online abgerufen am 14.03.2016] URL:
<https://www.flickr.com/photos/shanti/2190115732/>
- [130] **TABLICE LED, TELEBIMY** [Online abgerufen am 07.02.2016] URL:
<https://www.youtube.com/watch?v=dbXNyi8IO14>
- [131] **INARAY Design Group** [Online abgerufen am 22.02.2016] URL:
<http://www.inaray.com/led-lighting/the-amazing-benefits-of-led-lighting/>
- [132] **praeniteo GmbH** [Online abgerufen am 03.03.2016] URL:
<http://www.praeniteo.com/preisanzeigen-benzinpreisanzeigen.html>
- [133] **praeniteo GmbH** [Online abgerufen am 03.03.2016] URL:
<http://www.praeniteo.com/preisanzeigen-benzinpreisanzeigen.html>

Tabellenverzeichnis

[TAB1]	Raspberry Pi Modell B+ Datenblattübersicht <i>https://learn.adafruit.com/downloads/pdf/introducing-the-raspberry-pi-model-b-plus-plus-differences-vs-model-b.pdf</i>
[TAB2]	Distributionsübersicht <i>http://www.raspberry-pi-geek.de/Magazin/2013/05/Distributionen-fuer-den-Raspberry-Pi-im-Ueberblick</i>
[TAB3]	Vergleichsaufstellung Einplatinencomputer <i>http://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/06/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-2700x900.png</i>
[TAB4]	Problemauswertung Datenträger <i>https://www.elektronik-kompodium.de/sites/raspberry-pi/2101221.htm</i> <i>http://www.techradar.com/how-to/computing/how-to-fix-raspberry-pi-boot-problems-1310697</i> <i>http://techcrunch.com/2012/10/21/getting-started-with-the-raspberry-pi-is-not-as-easy-as-pie/</i>
[TAB5]	Problemauswertung Display <i>ebd.</i>
[TAB6]	Problemauswertung Stromversorgung <i>ebd.</i>
[TAB7]	Problemauswertung LED-Matrix <i>ebd.</i>

Abbildungsverzeichnis

- [ABB1] https://d3s5r33r268y59.cloudfront.net/05621/products/thumbs/2014-07-17T10:56:51.442Z-rsz_b-.jpg.855x570_q85_pad_rcrop.jpg
- [ABB2] <http://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/06/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-2700x900.png>
- [ABB3] <http://www.tacticalcode.de/2013/01/led-matrix-grundlagen-mit-multiplexing.html>
- [ABB4] <https://learn.adafruit.com/connecting-a-16x32-rgb-led-matrix-panel-to-a-raspberry-pi/you-will-need>
- [ABB5] <https://learn.adafruit.com/connecting-a-16x32-rgb-led-matrix-panel-to-a-raspberry-pi/you-will-need>
- [ABB6] *ebd.*
- [ABB7] *ebd.*
- [ABB8] <https://www.adafruit.com/products/2693>
<https://learn.adafruit.com/connecting-a-16x32-rgb-led-matrix-panel-to-a-raspberry-pi/you-will-need>
- [ABB9] <https://learn.adafruit.com/connecting-a-16x32-rgb-led-matrix-panel-to-a-raspberry-pi/you-will-need>
- [ABB10] <http://www.amazon.de/bao-Cat-Stecker-Netzwerkkabel-Gelb/dp/B00JD8TC1I>
- [ABB11] http://www.amazon.de/AmazonBasics-HL-007304-Hochgeschwindigkeits-HDMI-Kabel-HDMI-Standard-2-0/dp/B014I8SIJY/ref=lp_316968011_1_2/277-5238359-5712951?s=ce-accessories&ie=UTF8&qid=1458980486&sr=1-2
- [ABB12] Marko Mihaljevic © 2016
- [ABB13] Marko Mihaljevic © 2016
- [ABB14] <https://learn.adafruit.com/assets/23825>
- [ABB15] <https://learn.adafruit.com/assets/23920>
- [ABB16] Marko Mihaljevic © 2016
- [ABB17] [https://learn.adafruit.com/assets/15207\(Oben\)](https://learn.adafruit.com/assets/15207(Oben)), Marko Mihaljevic © 2016
(unten links sowie unten rechts)
- [ABB18] Marko Mihaljevic © 2016
- [ABB19] http://www.raspberrypi-spy.co.uk/wp-content/uploads/2015/03/sd_formatter_guide_01.png
<http://www.raspberrypi-spy.co.uk/2015/03/how-to-format-pi-sd-cards-using-sd-formatter/>

- [ABB20] http://www.raspberrypi-spy.co.uk/wp-content/uploads/2015/03/sd_formatter_guide_05.png x
- [ABB21] <http://nighttype.com/wp-content/uploads/2014/01/raspi-config.png>
- [ABB22] http://www.jann.cc/_images_big/rpi/raspberry_pi_boot-big.jpeg
- [ABB23] Marko Mihaljevic © 2016
- [ABB24] https://learn.adafruit.com/system/assets/assets/000/002/868/original/learn_raspberry_pi_terminal.png?1396788119
- [ABB25] Marko Mihaljevic © 2016
- [ABB26] <http://tothepc.com/img/2010/03/putty-client.png>
- [ABB27] https://www.tu-chemnitz.de/informatik/friz/windows_xp/putty/3.jpg
- [ABB28] https://raw.githubusercontent.com/adafruit/Adafruit-Pi-Finder/master/docs/rpi_bootstrap.png
- [ABB29] <http://blog.queststem.com/post/104307655891/quick-start-guide-gerts-passive-vga-adapter-666>
- [ABB30] <http://www.instructables.com/id/Stream-games-to-your-Raspberry-Pi-2/?ALLSTEPS>
- [ABB31] <http://www.thirdeyevis.com/images/3v-led-diagram.png>
- [ABB32] <http://blog.goo.ne.jp/toko0131/e/025bbebce409c3090ffa43018720d779>
- [ABB33] Marko Mihaljevic © 2016
- [ABB34] Marko Mihaljevic © 2016
- [ABB35] Marko Mihaljevic © 2016
- [ABB36] Marko Mihaljevic © 2016
- [ABB37] Marko Mihaljevic © 2016
- [ABB38] Marko Mihaljevic © 2016
- [ABB39] Marko Mihaljevic © 2016
- [ABB40] Marko Mihaljevic © 2016
- [ABB41] Marko Mihaljevic © 2016
- [ABB42] http://www.viaberlin.de/web/img/pj/1053000_2.jpg
- [ABB43] https://www.swarco.com/var/em_plain_site/storage/images/media/images/650-access-and-parking/swarco_solution_parking-001_320x160rgb/23780-1-eng-US/SWARCO_Solution_parking-001_320x160rgb_795x530px.jpg
- [ABB44] http://www.yatego.com/spitze-handy/p,4ca20f31028eb,403c63154b9d25_9,doener-kebab-led-leucht-reklame-anzeige-tafel
- [ABB45] <https://plus.google.com/+GlobalShishaLoungeOffenbach/videos>

Abkürzungsverzeichnis

ARM	Acorn RISC Machines (CPU-Architektur)
DHCP	Dynamic Host Configuraiton Protocol
DVI	Digital Visual Interface
GB	Gigabyte
GHz	Gigahertz
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
HDMI	High Definition Multimedia Interface
IP	Internetprotokoll
LAN	Local Area Network
LED	Light-Emitting Diode
MB	Megabyte
Mbit / s	Megabit pro Sekunde
MHz	Megahertz
PoP	Package-on-Package
SD	Secure Digital
SoC	System on a Chip
USB	Universal Serial Bus
WLAN	Wireless Local Area Network