

Timotheus Hell, BSc

Zeitrafferaufnahmen mit dem Raspberry Pi Zero

Ein Projekt für den Schulunterricht

MASTERARBEIT

zur Erlangung des akademischen Grades
Diplomingenieur

Masterstudium Telematik

eingereicht an der
Technischen Universität Graz

Betreuer
PD DI Dr. Martin Ebner

Institute of Interactive Systems and Data Science

Graz, April 2018

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Danke, Jana, Martin und Ricky.

Kurzfassung

Um mehr Schüler/innen für ein technisches Studium zu motivieren braucht es ein Mehr an begeisternden Projekten, die etwa im Rahmen des Informatik- oder des neuen Werkunterrichts umgesetzt werden können. Das in dieser Arbeit erarbeitete Projekt, Zeitraffer Video- und Tonaufnahmen automatisiert am Raspberry Pi (RPI) Zero zu erstellen, setzt dazu auf günstige, verfügbare Hardware, einfache Umsetzung und quelloffene Software.

Die Arbeit geht auf interdisziplinäre Aspekte ein, die helfen, um Schüler/innen mit verschiedenen Interessen für ein solches hardwarenahes, technisches Projekt zu begeistern: Die Entwicklung und Technik der Zeitrafferfotografie, Bild-, Video- und Audioformate, sowie rechtliche und künstlerische Aspekte. Es werden auch Erweiterungen präsentiert, die sich mit der RPI Plattform umsetzen lassen - etwa für eine Wetterstation.

Außerdem wird eine eigene Methode zur Aufnahme von natürlich klingenden Zeitraffer-Tonaufnahmen vorgestellt.

Technisch liegt der Anspruch vor allem in der optimalen Ausnutzung von besonders günstiger Hardware. Die Ansprüche werden im ersten Teil der Arbeit motiviert, der zweite Teil dokumentiert die Implementierung und der dritte Teil evaluiert das Projekt anhand der gesetzten Kriterien. Als Motiv wird dafür eine Baustelle gewählt, eine typische Anwendung für Zeitrafferaufnahmen die über den Einsatz in der Lehre hinaus auch generell interessant sein kann, um Bauvorhaben zu dokumentieren.

Das Projekt kann einfach und mit Hardwarekosten von etwa 50 Euro nachgebaut werden, in der Arbeit sind alle notwendigen Schritte beschrieben, alle Software ist frei verfügbar und wird zusammen mit dieser Arbeit veröffentlicht. Die Arbeit, entwickelte und verwendete Software stellen damit eine freie Bildungsressource und freie, offene Software dar.

Abstract

To encourage more students to pursue a STEM degree, exciting projects for school subjects such as computer science are needed. The project developed within the scope of this work, time-lapse video and audio recordings automated on the Raspberry Pi (RPI) Zero, relies on affordable, available hardware, easy implementation and Open Source software.

The work addresses interdisciplinary aspects that are helpful in enthusing students with diverse interests in a hardware-related, technical project: the history, development and technique of time-lapse photography, image, video and audio formats, as well as legal and artistic aspects are all investigated in the scope of this thesis. It also presents extensions that can be implemented on the RPI platform, e.g. a weather station.

In addition, a method for recording natural-sounding time-lapse sound recordings is presented.

On a technical level, the thesis' intent is the optimal utilization of affordable hardware. The goals are motivated in the first part of the work, the second part documents the implementation and the third part evaluates the project based on the set criteria. As an example, we choose a construction site, a typical application for time-lapse photography which is interesting not only in the scope of teaching but in general to showcase new construction projects.

The project can be copied easily and with hardware costs of about 50 Euro. All the necessary steps are detailed in this thesis, all software is freely available and will be published alongside this work. The thesis and developed and used software thereby represent an open educational resource and open and free software.

Inhaltsverzeichnis

1. Einleitung	5
1.1. Zielsetzung	6
1.2. Struktur der Arbeit	6
I. Grundlagen	7
2. Zeitraffer	8
2.1. Geschichte	10
2.2. Stand der Technik	12
2.3. Audio-Zeitraffer	16
3. Rechtliche Aspekte	18
3.1. Urheberrecht, FOSS, OER Lizenzen	18
3.2. Persönlichkeitsrecht und Privatsphäre	20
3.3. Recht am eigenen Bild und Urheberrecht	22
3.4. Bildaufnahme und Videoüberwachung	22
3.5. Strafrecht - Strafgesetzbuch	24
3.6. Rechtsprechung	24
3.7. Zusammenfassung	26
4. Wahl der Hard- und Software	27
4.1. Hardware	27
4.1.1. Computer	27
4.1.2. Kamera	28
4.1.3. Unterbrechungsfreie Stromversorgung (USV)	30
4.1.4. Gehäuse	30
4.1.5. Akku	32
4.1.6. Mikrofon	35
4.1.7. Speicherkarte	35
4.2. Software	37
4.2.1. Betriebssystem	37
4.2.2. Skriptsprache	39
4.2.3. Kamera	39

Inhaltsverzeichnis

4.2.4. Video	39
4.2.5. Audio	40
4.3. Bild-, Audio- und Videoformate	40
4.3.1. Bild	40
4.3.2. Audio	41
4.3.3. Video	43
4.3.4. Container	44
5. Zusammenfassung	45
II. Umsetzung	46
6. Konfiguration	47
6.1. Einrichtung, Aktualisierungen, Installation von Softwarepaketen	47
6.2. BIOS-Konfiguration	48
6.3. Auslagerungsdatei	49
6.4. Wireless Lan	50
6.5. Mikrofone einrichten	50
7. Zeitrafferaufnahmen am RPI	51
7.1. Verschiedene Lösungsansätze	51
7.2. Aufnahme der Einzelbilder	55
7.3. Zeitraffertonaufnahmen	57
7.4. Erstellung des Videos	61
7.5. Logging	63
7.6. Erweiterungen	65
7.7. Zusammenfassung	74
III. Analyse	75
8. Ergebnisse	76
8.1. Das Motiv	76
8.2. Aufnahmen	77
8.3. Anwendungsbeispiele	77
8.4. Robustheit	84
8.5. Hardwarekosten	84
8.6. Aufwandsabschätzung für den Nachbau	87
9. Zusammenfassung	88
Literatur	94

Abkürzungsverzeichnis

BIOS	Basic Input/Output System
CC	Creative Commons
CRF	Constant Rate Factor
DSI	Display Serial Interface
FIFO	First In – First Out
FOSS	Free and Open Source Software
GPIO	General Purpose Input/Output
GPL	GNU General Public License
HDMI	High Definition Multimedia Interface
HDR	High Dynamic Range
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
LiPo	Lithium-Polymer-Akkumulator
MPPT	Maximum Power Point Tracking
OER	Open Educational Resources
PCM	Puls Code Modulation
PID	Process identifier
POE	Power over Ethernet
PWM	Pulsweitenmodulation
THD	Total Harmonic Distortion
TSM	Time Scale Modification
USV	Unterbrechungsfreie Stromversorgung
VWA	Vorwissenschaftliche Arbeit
WSOLA	Waveform Similarity and Overlap Add

Tabellenverzeichnis

2.1. Einige RPI Modelle	13
8.1. Hardware Einkaufsliste	86

Abbildungsverzeichnis

2.1. Muybridges Aufnahme "The Horse in Motion"	10
2.2. Time-lapse Apparat Connected to Camera	11
2.3. RPI 0 und 2	14
2.4. RPI 3+	15
4.1. RPI Kamera	29
4.2. Kondensatoren zur RPI Energieversorgung	31
4.3. Gehäuse	33
4.4. RPI LiPo Akku	34
4.5. RPI Mikrofon	36
7.1. Waveform Filter am Audiosignal	59
7.2. Der Splice Befehl	60
7.3. RPI 2 und Sensoren	67
7.4. MPPT-Modul	69
7.5. Feinstaubsensor	70
7.6. HDR Aufnahmen	72
7.7. Tilt-und-Shift Aufnahme	73
8.1. Kamera von Außen	77
8.2. Bebauungsplan 06.23.0	78
8.3. 18-04-19, 8:33:15, Betonpumpe	78
8.4. 18-04-01, 22:59:00, Selfie	79
8.5. 18-04-16, 22:42:00, Regentropfen	79
8.6. RPI Zero, enfused	80
8.7. RPI Zero, luminanceHDR	80
8.8. RPI 2 mit Sensordaten, Blitz	81
8.9. RPI 2 mit Sensordaten, Blitzbeleuchtung	81
8.10. RPI 2 mit Sensordaten, sonniger Sommertag	82
8.11. RPI 2 mit Sensordaten, heftiger Sturm	82
8.12. Logs	85
8.13. Temperaturverlauf Graz	86

Listings

4.1. Benchmark SD-Karte vs Ramdisk	38
4.2. Benchmark jpg versus png	41
7.1. Anlegen der Pipe	53
7.2. Start von ffmpeg	53
7.3. Übergabe der Bilder via Pipe	54
7.4. Fehlerbehandlung	55
7.5. Aufnahme eines Fotos mit raspistill	56
7.6. Aufnahme eines Fotos mit raspistill - Kernbefehl	56
7.7. Abfrage PIPESTATUS	57
7.8. Fehlerbehandlung	57
7.9. Tonaufnahme	58
7.10. Filtern des Audiosignals	58
7.11. Splice mit sox	60
7.12. Zurechtschneiden der Tonspur	61
7.13. Zusammenführen von Video- und Tonspur	62
7.14. Start von ffmpeg, Audio und Video in Container	62
7.15. Pipe zu ffmpeg	63
7.16. Auslesen der USV	64
7.17. Logging	64
7.18. Komposition eines Bildes mit imagemagick	66
7.19. Tilt-und-Shift Filter mit imagemagick, adaptiert von http://www.imagemagick.org/Usage/photos/ , abgerufen 18-04-25	73

1. Einleitung

Einplatinencomputer gewinnen Dank des Erfolges des Raspberry Pi (RPI) gerade an Bedeutung für viele Anwendungsbereiche. Einerseits ermöglichen sie kostengünstige Lösungen für Projekte etwa in der Kunst, Industrie oder der Medizin, andererseits wird die Plattform als Bildungswerkzeug eingesetzt, so wie es auch der Intention des RPI Projektes entspricht (Johnston und Cox, 2017). Dazu finden sich zahlreiche auf dem RPI basierende Projekte¹ ebenso wie Arbeiten, die sich mit den pädagogischen Aspekten von Making im Unterricht am Beispiel des RPI im Schulunterricht befassen - beispielsweise die gleichnamige Diplomarbeit (Müller, 2016). Alleinstellungsmerkmal des RPI gegenüber anderen Produkten ist die Kombination ein „vollwertiger Computer“² zu sein mit dem niedrigen Preis und der guten Verfügbarkeit einerseits und der großen und aktiven Community andererseits: Im RPI Forum alleine befinden sich mit Stand von April 2018 über 1,2 Millionen Beiträge zu knapp 200.000 Themen von gut ebensovielen Benutzer/innen³. Dazu kommt ein vielfältiges Angebot von Hardware, die speziell für die Plattform ausgesucht ist, und mit Anleitungen versehen angeboten wird⁴.

Damit ist die RPI Plattform gut geeignet, um darauf basierende Projekte für einen Einsatz im Unterricht zu gestalten. Ein passender Rahmen dazu tut sich aktuell an der TU Graz auf, die in das mit 2018 neue Unterrichtsfach „Technische und Textile Gestaltung“ im Rahmen des Kooperationsstudiums Lehramt Sekundarstufe Allgemeinbildung einsteigen wird. Dieses Studium soll den angehenden Lehrer/innen nun erstmals den "Makerspace als Ort der Bildung und Wissensvermittlung, des kreativen digitalen Gestaltens sowie als Kristallisationspunkt für Innovationen und Start-Ups"⁵ vermitteln. Im Wege über die Lehrer/innen, die im Unterricht Projekte durchführen, die Schüler/innen zu einem technischen Studium motivieren und gut vorbereiten, wird auch ein wichtiger Beitrag für den Wirtschaftsstandort geleistet: Wie Initiativen wie das Software and Data Council Styria⁶ aufzeigen, werden dringend mehr Absolvent/innen der informatiknahen Studien benötigt. Beispiele für spannende und (finanziell) im

¹Beispielsweise hier zu finden: ieeexplore.ieee.org/search/searchresult.jsp?queryText=raspberry&newsearch=true, abgerufen 18-04-23

²siehe z. B. www.raspberrypi.org/magpi/raspberry-pi-desktop-pc/, abgerufen 18-04-23

³www.raspberrypi.org/forums/, abgerufen 18-04-23

⁴zu finden beispielsweise bei Adafruit, www.adafruit.com/category/105, abgerufen 18-04-23

⁵vgl. Entwurf Curriculum BA Lehramt Curriculum 2017, Fassung 2018

⁶dazu entsteht gerade die Website www.sdc-styria.at, abgerufen 18-04-23

1. Einleitung

Rahmen des Unterrichts umsetzbare Projekte basierend auf der RPI Plattform gewinnen auf diesem Weg hohe gesellschaftliche Relevanz. Dabei erlauben es interdisziplinäre Projekte, neue Zielgruppen für technische Studien zu erschließen - etwa indem Aspekte aus dem Bild-, Video- und Tonmedienbereich die Beschäftigung mit den technischen Voraussetzungen motivieren. Diese Masterarbeit beschreibt ein solches Projekt.

1.1. Zielsetzung

Basierend auf dem RPI soll eine Zeitrafferkamera entwickelt werden. Ziel ist eine optimale Ausnutzung der Kapazitäten dieser Plattform basierend auf quelloffener Software und die Zurverfügungstellung eines Projektes mit audiovisuell ansprechenden Videos als Ergebnis, das fortan im Unterricht eingesetzt werden kann.⁷ Die Funktion soll anhand der Aufnahme einer Baustelle getestet werden: Das ist eine typische Anwendung, die helfen kann, Anrainer/innen und Stadtbewohner/innen für ein Bauprojekt, mit dem häufig auch Lärm- und Staubemissionen einhergehen, zu gewinnen.

Die Formulierung der Forschungsfrage dazu lautet: Welches Ergebnis kann basierend auf den Einschränkungen der Hardware des Raspberry Pi Zero für eine robuste, kostengünstige, automatisierte Zeitrafferkamera mit dem Einsatz quelloffener Software erreicht werden?

1.2. Struktur der Arbeit

Die Arbeit ist in drei Teile gegliedert: Im ersten Teil werden gängige Wege um Zeitrafferaufnahmen zu erstellen und verschiedene Automatisierungsansätze beschrieben, die Ziele für ein 'optimales' Ergebnis erarbeitet, rechtliche Fragen beleuchtet und die Wahl der Hard- und Software motiviert.

Im Umsetzungsteil sind verschiedene Varianten für einzelne Problemstellungen dargestellt, wobei jeweils die geeignetste gemäß der erarbeiteten Kriterien schließlich zur Anwendung kommt. Außerdem werden Erweiterungsmöglichkeiten beschrieben und getestet, die auf leistungsstärkeren Varianten des RPI eingesetzt werden können.

Der dritte Teil der Arbeit analysiert das Gesamtergebnis hinsichtlich der im ersten Teil erarbeiteten Anforderungen und schließt die Arbeit mit der Diskussion der Ergebnisse ab.

⁷Diese Arbeit und aller zum Nachbau des Gerätes erforderliche und im Rahmen der Arbeit erstellte Programmcode stehen dazu gemäß der Creative Commons BY-SA 3.0 AT Lizenz zur freien Verwendung.

Teil I.

Grundlagen

Einleitung

Im ersten Teil dieser Arbeit wird in die Zeitrafferfotografie eingeführt und die Geschichte und (aktuelle) Entwicklung der Technik umrissen. Ein kurzer Exkurs führt durch die Geschichte des RPI und ist wichtig, um später die Entscheidung für diese Plattform motivieren zu können. Auch den Möglichkeiten, Ton im Zeitraffer aufzunehmen und den rechtlichen Aspekten sind jeweils ein eigenes Kapitel gewidmet. Den Abschluss des ersten Teils bildet eine Analyse verschiedener Hard- und Softwareoptionen: Jeweils die Option, die sich als am besten geeignet abzeichnet, wird dann in Teil 2 im Projekt zum Einsatz gebracht.

2. Zeitraffer

Das Gehirn interpretiert Änderungen zwischen rasch aufeinander folgend wiedergegebenen Bildern mit hinreichend ähnlichem Inhalt als Bewegungen. Videoaufzeichnungen nutzen diesen Effekt, um fließende Bewegungen durch eine Abfolge aufeinanderfolgender Einzelbilder anzunähern. Dabei ist je nach Videostandard eine Bildrate von zumindest 20 Bildern pro Sekunde und mehr üblich¹, um Bewegungen fließend erscheinen zu lassen. Liegt das Ziel in einer möglichst originalgetreuen Wiedergabe der ursprünglichen Szene, so muss die Frequenz der Aufnahme also einerseits möglichst hoch sein und andererseits genau der Wiedergabefrequenz entsprechen. Neben der Bildrate, also der zeitlichen Auflösung, muss auch die räumliche Auflösung der Einzelbilder ausreichend hoch sein, damit alle wünschenswerten Details festgehalten werden. Idealerweise liegen also sowohl die zeitliche als auch die räumliche Auflösung der Einzelbilder über dem Auflösungsvermögen des menschlichen Auges. Das ist bei einer Bildrate von etwa 60 Bildern pro Sekunde und einer hinreichend großen Entfernung und räumlichen Auflösung, so dass einzelne Bildpunkte mit der Winkelsehschärfe des

¹bestes Beispiel sind die beiden analogen Fernsehstandards PAL und NTSC, die mit 50 beziehungsweise 60 Halbbildern pro Sekunde arbeiten

2. Zeitraffer

menschlichen Auges im Bereich von 25 Winkelsekunden bis einer Winkelminute nicht mehr aufgelöst werden können der Fall. Bei einem Abstand von 1 Meter sind bei dieser Winkelsehschärfe von 1" zwei Bildpunkte die 0,3 mm voneinander entfernt sind also noch auszumachen (Schramm, 2001). Ein Bild mit 1920 mal 1080 Bildpunkten (Full HD) hätte bei dieser Auflösung Abmessungen von etwa 58 mal 32 Zentimeter, was etwa einem 26 Zoll Computerbildschirm entspricht. Eine Änderung der räumlichen Auflösung in der Darstellung führt zu einer Änderung der Größe der dargestellten Objekte. Wird die zeitliche Auflösung verändert, so ändert sich die Filmlänge und es kommt zu Zeitlupe (Einzelbilder werden mit einer niedrigeren Bildrate wiedergegeben als aufgenommen) oder einem Zeitraffer (Einzelbilder werden mit einer höheren Bildrate wiedergegeben als aufgenommen). Mit

$$n = \text{Bildanzahl}[1]$$

$$l = \text{Filmdauer}[s]$$

$$r = \text{Bildrate}[1/s]$$

$$d = \text{Aufnahmedauer}[s]$$

$$i = \text{Intervall}[s]$$

ergeben sich die Zusammenhänge bei einer Zeitrafferaufnahme als

$$n = r * l \tag{2.1}$$

und entsprechend

$$n = d/i \tag{2.2}$$

Im Fall des Zeitraffers, dem Gegenstück zu einer Verkleinerung der räumlichen Auflösung und damit der Größe der Darstellung, steigt mit der höheren Bildrate auch die Qualität der Darstellung. Im Fall der Zeitlupe muss die Bildrate bei der Aufnahme um denselben Faktor höher sein, um den die Zeitdehnung erfolgen soll, um eine Darstellung mit den zumindest 20 notwendigen Bildern pro Sekunde erreichen zu können. Beide Varianten erlauben Phänomene und Vorgänge erkennbar zu machen, die sich in für das menschliche Auge zu langen oder zu kurzen Intervallen vollziehen. Damit sind Zeitrafferaufnahmen auch wissenschaftlich relevant: sie ermöglichen beispielsweise, das Paarungsverhalten der Krähenscharbe (Harris, 1982), mikroskopische Abläufe in Zellen (Landecker, 2009), Baufehler (Fox u. a., 2015) oder die Bewegung von Gletschern in den Anden (Carvallo u. a., 2017) zu analysieren. Im letztgenannten Beispiel von 2017 wird auch bereits ein RPI eingesetzt, allerdings ein RPI 3 und lediglich zur Übertragung der einzelnen Bilder, nicht zur Erstellung eines Videos. Da in vielerlei Hinsicht Zeitlupenaufnahmen höhere Anforderungen Stellen (insbesondere betreffend

2. Zeitraffer

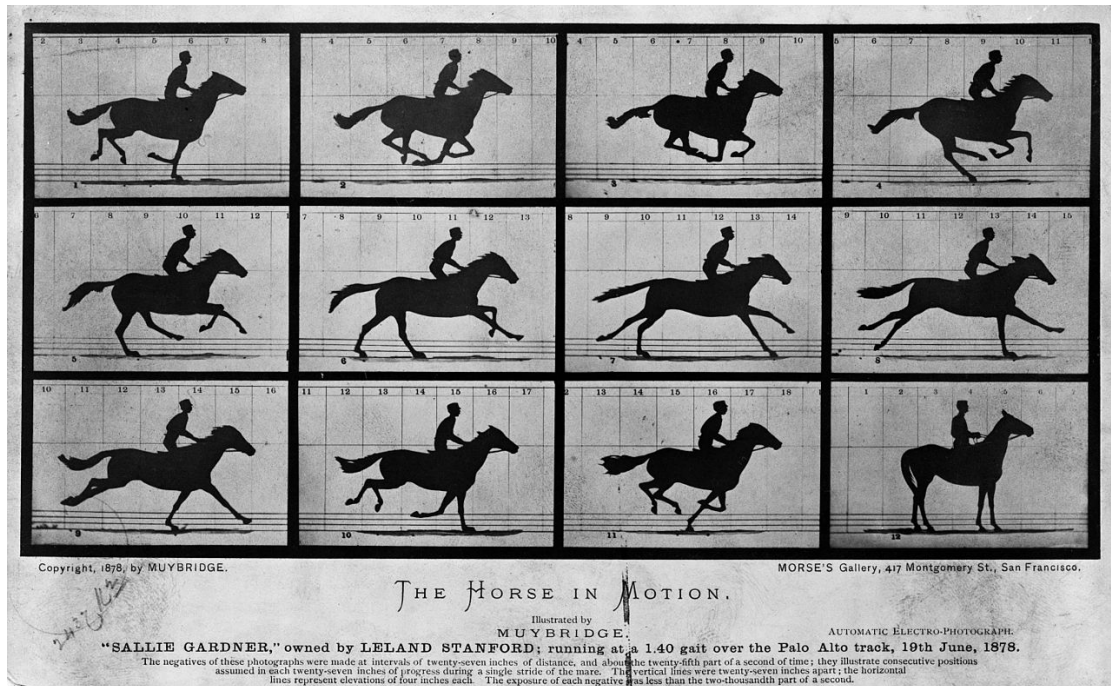


Abbildung 2.1.: Muybridges berühmte Aufnahme "The Horse in Motion", von https://en.wikipedia.org/wiki/File:The_Horse_in_Motion_high_res.jpg, Quelle: Library of Congress Prints and Photographs Division, 19 June 1878, Autor: Eadweard Muybridge

die Kameraoptik für eine kurze Belichtungszeit, Speicher, Verarbeitungsgeschwindigkeit) sind Zeitrafferaufnahmen eine wesentlich geeignetere Anwendung für Hardware mit eng begrenzten Kapazitäten.

Aus den dargestellten Gründen ist für das Projekt ein Video mit einer FullHD Auflösung und 60 Bildern pro Sekunde (kurz 1080p60) ein anzustrebendes Ergebnis.

2.1. Geschichte

Eine knappe Zusammenfassung über die knapp 150-jährige Geschichte der Zeitrafferaufnahme gibt Chris Weston im Buch *Spanning Time* (Weston, 2015): Um die Frage zu klären, ob im Galopp alle vier Beine eines Pferdes den Bodenkontakt verlieren, engagierte der einstige Gouverneur Kaliforniens Leland Stanford im Jahr 1872 den Fotografen Eadweard Muybridge. Neben der Antwort² entsprangen dieser Arbeit auch die ersten time-lapse Aufnahmen³, dargestellt in Abbildung 2.1.

²Ja.

³Hier scheint die übliche Übersetzung Zeitraffer noch unpassend, da sich der zeitraffende Effekt erst in einer schnelleren Wiedergabe einstellt.

2. Zeitraffer

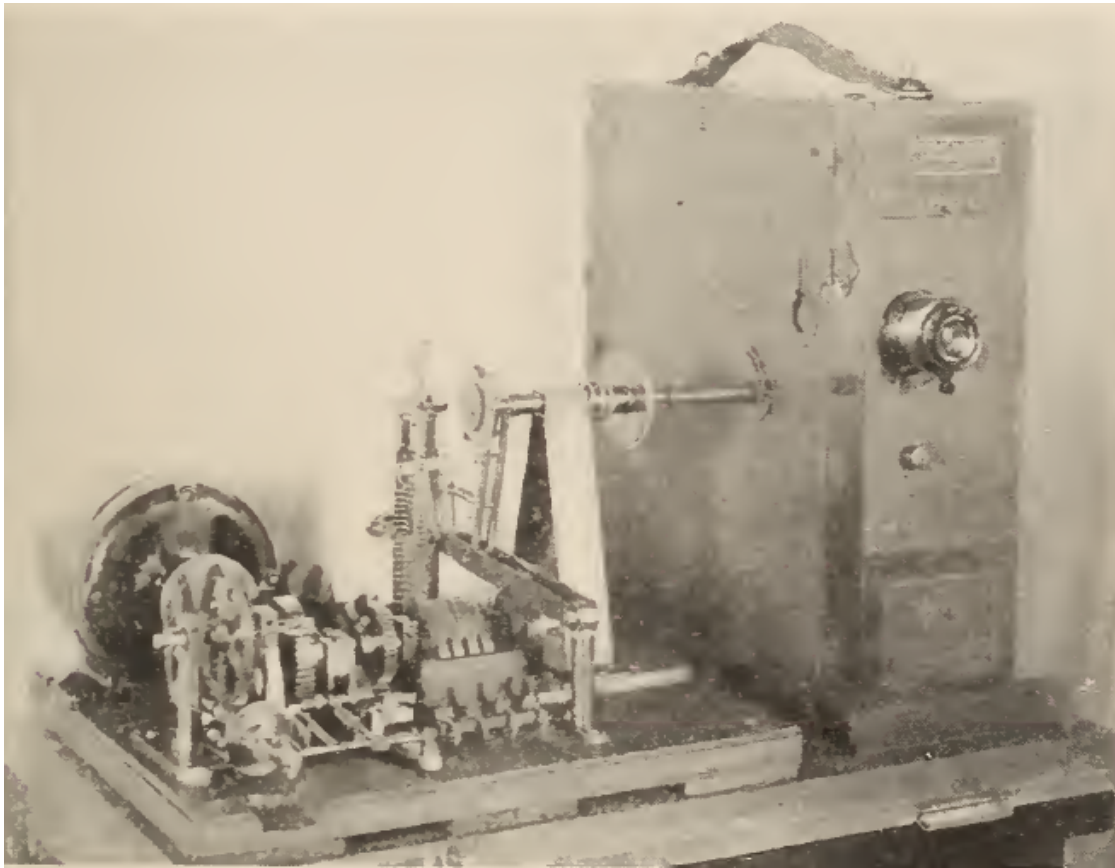


Abbildung 2.2.: Time-lapse Apparatus Connected to Camera, aus Green, 1926, S 149

Eine der ersten Apparaturen zur Automatisierung der Aufnahmen zur Erstellung von Zeitraffervideos entwickelte in den 1920ern George Georgens für die Abteilung für Bewegtbilder des Landwirtschaftsministeriums der Vereinigten Staaten (Green, 1926). Damit sollten natürliche Vorgänge wie Samenkeimung, Schimmelbildung, Pflanzenwachstum, die Entwicklung von Pflanzen- und Baumkrankheiten und dergleichen sichtbar gemacht werden (Green, 1926, S. 147). Der Apparat ist in Abbildung 2.2 gezeigt. Hier zeigen sich schon Anforderungen an die Robustheit der eingesetzten Methodik: "When the current supply is resumed, it may catch the clock switch entirely out of time with the rest of the machine. This, however, takes care of itself, and automatically the proper relationship between the various parts is resumed in the next cycle. The worst that can happen is that a single frame may be spoiled, but even this is unlikely. It will simply pick up where it left off and go on running as if nothing had happened." (Green, 1926, S. 151)

Zumindest dieses Niveau an Zuverlässigkeit, das vor über 90 Jahren erreicht werden konnte, sollte auch die Mindestanforderung an eine heutige Entwicklung darstellen.

Erst in den 1930ern stand dann bei den Arbeiten von John Ott der Zeitrafferaspekt im

2. Zeitraffer

Vordergrund. Er nutzte seine Aufnahmen um das Wachsen von Pflanzen zu visualisieren (Weston, 2015). Seine weltbekannten Aufnahmen finden sich beispielsweise im Dokumentarfilm *Disney's Secrets of Life* von 1956. Je nach Motiv sind offensichtlich unterschiedliche Aufnahmeintervalle geeignet: Um das Wachstum eines Baumes zu zeigen, sollten wenige Bilder im Monat genügen, der Fortschritt eines Bauvorhabens ist - je nach Fokus - mit einem Bild am Tag (etwa um zu zeigen, wie ein Gebäude in die Höhe wächst) oder auch mehreren Bildern pro Minute (um z.B den Aufbau eines Krans zu zeigen) gut darzustellen. Um ausreichend Flexibilität zu gewähren, ist für das hier entwickelte Projekt anzustreben, dass mehrere Bilder pro Minute aufgenommen und verarbeitet werden können.

2.2. Stand der Technik

Für die Erstellung von Zeitrafferaufnahmen existieren heute Intervalometer, die in einstellbaren Zeitabständen Impulse abgeben und damit eine Aufnahme auslösen. Digitalkameras integrieren diese Funktion mitunter auch⁴. Im professionellen Bereich werden üblicherweise SLR Kameras zusammen mit Intervalometern genutzt.⁵ Am Markt existieren auch einzelne integrierte Lösungen, also Kameras, die speziell für die Zeitrafferaufnahme ausgelegt sind⁶. Diverse Action-Camcorder und natürlich Smartphones unterstützen ebenfalls Zeitrafferaufnahmen. Zumindest in der Forschung gibt es auch schon Ansätze ohne (eigene) Kamera Aufnahmen aus im Internet verfügbaren Fotos zu erstellen (Martin-Brualla, Gallup und Seitz, 2015). Beispiele für aktuelle Anleitungen, wie Zeitrafferaufnahmen erstellt werden können liefern etwa (Weston, 2015) und (Higgins, 2017).

Aber nicht nur in der Forschung, auch in Kunst und Kultur ist die Zeitrafferaufnahme lange schon schon als Stilmittel angekommen und auch technisch weiterentwickelt worden: Im Film *Baraka* (Fricke, 1992, etwa bei 42:10) wird als einem der ersten Filme eine Kamerabewegung im Zeitraffer eingesetzt. Der Film zeigt damit das hektische Treiben an einer Kreuzung einer amerikanischen Großstadt zu dem die langsame Kamerabewegung einen Kontrapunkt setzt. Inzwischen gibt es viele Beispiele für Aufnahmen mit dieser auch Hyperlapse genannten Technik.

Der rasante Fortschritt bei digitalen Fotokameras hat nicht nur zur Verbreitung als Hobby geführt⁷, sondern ermöglicht auch weitere grundlegendere technische Fortschritte,

⁴siehe z.B. <https://www.prophoto-online.de/fototipps/Intervalometer-Zeitrafferaufnahmen-10006116>, abgerufen 18-04-23

⁵in diesem Video wird die Technik von Ben Canales sehr schön gezeigt <http://watch.opb.org/video/2288686259/>, abgerufen 18-04-23

⁶für eine Liste siehe z.B. <https://geizhals.at/?fs=zeitrafferkamera>, abgerufen 18-04-23

⁷siehe z. B. die vielen Beispiele unter <https://www.reddit.com/r/timelapse/>, abgerufen 18-04-23

2. Zeitraffer

Tabelle 2.1.: Einige RPI Modelle

Modell	RPI Pi 3 B+	RPI Pi Zero W	RPI Pi 2 B
Erschienen	2018	2017	2015
Preis	35 USD	10 USD	35 USD
# Kerne	4	1	4
CPU	BCM2837Bo	BCM2835	BCM2836
Architektur	64 Bit	32 Bit	32 Bit
Takt	1.4 GHz	1 GHz	0.9 GHz
Arbeitsspeicher	1 GB	512 MB	1 GB
Leistung	5.7 W	1.75 W	4.1 W

wie etwa die Aufnahme von High Dynamic Range (HDR) Zeitrafferfilmen, von denen erste Beispiele vor etwa zehn Jahren zu finden sind, z. B. Silicon Valley Timelapse von Nate North (North, 2008).

Im selben Jahr 2008 beginnt auch die Geschichte des RPI mit der Zusammenstellung des Raspberry Pi board of trustees und im darauffolgenden Jahr der Gründung der Raspberry Pi Foundation⁸. Das Ziel der in Großbritannien als Wohltätigkeitsorganisation eingetragenen Stiftung ist, nach eigener Darstellung inspiriert vom BBC Acorn aus den 1980er Jahren, die Förderung der Informatikbildung. Hierin findet sich auch immer noch die überwiegende Anwendung (Johnston und Cox, 2017).

Anfang 2015 waren bereits 5 Millionen RPI verkauft⁹, ein gutes Jahr später 11 Millionen¹⁰ und inzwischen (Stand März 2018) sind es 19 Millionen¹¹, was für einen breitenwirksamen Effekt des Konzeptes spricht.

Es existieren zahlreiche Modellvarianten. Die für das in dieser Arbeit vorgestellte Projekt verwendeten sind: RPI Zero W und RPI 2 Model B. Die aktuell leistungsstärkste Variante ist der RPI 3+ Model B. Die Abbildungen 2.3 und 2.4¹² zeigen jeweils ein Bild, Tabelle 8.1 der relevanten Eigenschaften: Erscheinungsjahr, Preis, Kerne, CPU-Typ, Architektur, Takt, Arbeitsspeicher, maximale Leistungsaufnahme¹³.

Zur Aufnahme von Fotos für Zeitraffervideos mit dem RPI gibt es etliche Anleitungen, von sehr simplen Varianten wie auf der Seite des Herstellers¹⁴, über etwas ausgefeiltere

⁸<http://apps.charitycommission.gov.uk/Showcharity/RegisterOfCharities/CharityFramework.aspx?RegisteredCharityNumber=1129409&SubsidiaryNumber=0>

⁹www.theguardian.com/technology/2015/feb/18/raspberry-pi-becomes-best-selling-british-computer

¹⁰www.raspberrypi.org/magpi/issues/53/

¹¹www.raspberrypi.org/blog/raspberry-pi-3-model-b-plus-sale-now-35/

¹²Die zusätzlich abgebildete TU-Karte dient auch in vielen weiteren Fotos zum Größenvergleich)

¹³Herstellerangaben, aus <https://www.raspberrypi.org/products/>, abgerufen 18-04-23

¹⁴raspberrypi.org/documentation/usage/camera/raspicam/timelapse.md, abgerufen 18-04-23

2. Zeitraffer

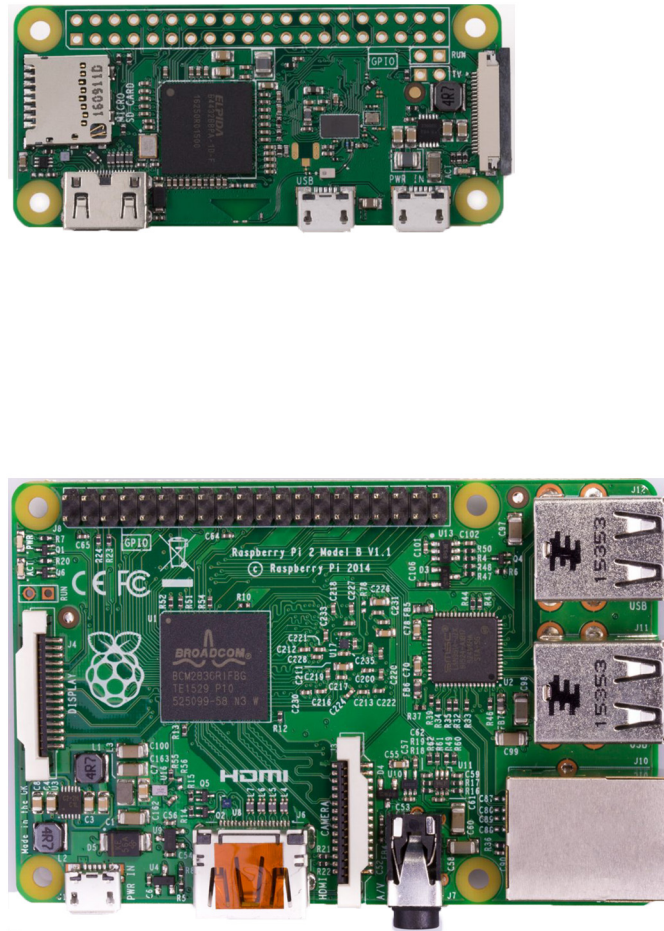


Abbildung 2.3.: RPI 0 und 2, Originalgröße (bei Druck in DIN A4), unter Verwendung der Fotos von <https://www.raspberrypi.org>.

2. Zeitraffer

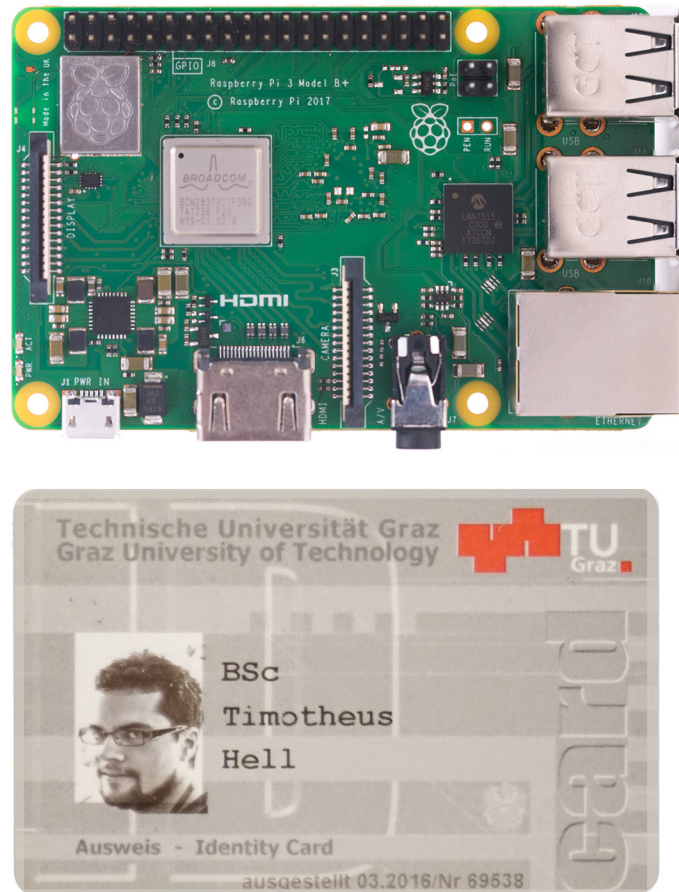


Abbildung 2.4.: RPI 3+ und TU-Card (ISO 7816 ID-1 Standardgröße) zum Größenvergleich, Originalgröße, unter Verwendung der Fotos von <https://www.raspberrypi.org>.

2. Zeitraffer

bis hin zu solchen die mit viel Overhead (Docker) die Aufnahmen automatisieren¹⁵, oder auch schon den RPI Zero W aufgrund seines Wlan verwenden¹⁶.

2.3. Audio-Zeitraffer

Um einen guten Gesamteindruck einer aufgenommenen Szene zu vermitteln, ist es wünschenswert wenn das Video auch über eine passende Tonspur verfügt. Üblicherweise werden für Zeitrafferaufnahmen Instrumentalstücke als Hintergrundmusik verwendet¹⁷. Ein einfaches Beschleunigen der Tonspur führt, anders als im Fall von Video, nicht zum erwünschten Ziel einer als "natürlich" empfundenen Wiedergabe. Um zu illustrieren, warum das so ist, wird im Folgenden auf das menschliche Hören im Vergleich zur technischen Umsetzung über ein Mikrofon eingegangen.

Was wir als "Hören" bezeichnen, sind Druckwellen, die sich in Frequenz und Amplitude unterscheiden und beliebig überlagern. Dabei nehmen wir eine Veränderung der Frequenz als eine Veränderung der Tonhöhe und eine Veränderung der Amplitude als eine Veränderung der Lautstärke wahr. Zu einem bestimmten Abtastzeitpunkt liegt ein bestimmter Druck am Ohr an, führt zu einer entsprechenden Verformung der Membran des Trommelfelles, einer mechanischen Weitergabe des Impulses in die Cochlea und schließlich der Übersetzung in einen Nervenimpuls (Zeller, 2009, S. 151ff). Auf vergleichbare Art wird Schall auch in einem Wandler - dem Mikrofon - in analog oder digital speicherbare elektrische Impulse gewandelt. Auch zur Reproduktion von Schallereignissen wird eine Membran verwendet: Die gespeicherten Impulse werden in derselben Folge und Geschwindigkeit wie bei der Aufnahme, meist nach einer elektrischen Verstärkung des Signals, an eine Spule gelegt und setzen die Membran und damit die Luft in Bewegung. Wird das Signal nun einfach in kürzerer Zeit abgespielt, so ändern sich damit genau jene Frequenzen, die neben der Amplitude, das zu reproduzierende Schallereignis ausmachen. Es kommt damit zu einer Änderung der Tonhöhe. Es existieren zwar in der Audiotechnik schon lange Zeit Methoden zur Tonhöhenkorrektur, doch stoßen diese bei starkem Zeitraffer an ihre Grenzen.

Es gibt auch einzelne Vorarbeiten bezüglich eines anderen Ansatzes zu einem Audiozeitraffer, wobei mit dem Aneinanderreihen von in Frequenzbestandteile zersetzter Musikstücke (DuBois, 2011)¹⁸ andere Ziele verfolgt werden. Im Internet finden sich bei

¹⁵blog.alexellis.io/raspberry-pi-timelapse/, abgerufen 18-04-23

¹⁶www.jeffgeerling.com/blog/2017/raspberry-pi-zero-w-headless-time-lapse-camera, abgerufen 18-04-23

¹⁷Viele Beispiele finden sich etwa auf youtube - https://www.youtube.com/results?search_query=time+lapse, abgerufen 18-04-23

¹⁸Ein Beispiel dieser Arbeiten kann unter <https://cantaloupemusic.com/albums/timelapse> gehört werden, abgerufen 18-04-23

2. Zeitraffer

Andrew Spitz¹⁹ und Rene Coronado²⁰ bereits bessere Beispiele: Es wird nicht laufend, sondern nur in einem bestimmten Intervall eine kurze Zeit Ton aufgenommen, so abgestimmt, dass die dann ohne Pause aneinander gereihten Aufnahmen insgesamt der Länge des Videos entsprechen. Deutlich hörbar ist bei dieser trivialen Herangehensweise ein "Knacken", da der Übergang zwischen den einzelnen Aufnahmen ruckartig erfolgt.

Erstrebenswert ist für das Projekt eine Audioaufnahme, die die Geräuschkulisse erkennbar wiedergibt, ohne dass Artefakte eingebracht werden. Die Audioaufnahmen können also nicht einfach aneinandergesetzt werden, sondern müssen beispielsweise mit einem fade-in und fade-out versehen überlagert werden.

Im Bereich der Audiosignalverarbeitung wird die dahinterliegende Aufgabenstellung, nämlich das Tempo eines Audiosignals ohne Änderung der Tonhöhe (beziehungsweise generell unabhängig voneinander Tonhöhe (pitch), Tempo und Geschwindigkeit (speed) eines Signals zu ändern) natürlich schon länger beforscht. Die Forschungsfrage der Time scale modification (TSM) führt bis in die 1950er Jahre, damals eingesetzt, um Redundanz im Gesprochenen auszunutzen (Fairbanks, Everitt und Jaeger, 1954). Heute existieren dazu viele Ansätze, eine der gängigen Methoden dazu ist die OLA (Overlap and Add) Technik - die der hier gewählten Technik sehr ähnlich ist: Die Grundidee des Algorithmus besteht darin, kleine Segmente aus dem Eingangssignal auszuschneiden und sie durch Überblenden (cross-fade) von einem Segment zum nächsten zu verketten, um die gewünschte Tempoänderung zu erreichen. Weiterentwicklungen dieser Grundidee sind immer noch in Verwendung, beispielsweise WSOLA (Wave Similarity Overlap and Add), das Artefakte, die aus den nicht zusammenpassenden Phasenlagen der aneinandergereihten Signale entstehen, vermeidet (Verhelst und Roelands, 1993). WSOLA sucht dazu innerhalb eines kleinen Intervalls nach Ähnlichkeiten in den zu überblendenden Signalen und nutzt diese Stellen, um das nächste Stück passgenau einzufügen. Eine gute Übersicht über TSM inklusive WSOLA für Musik gibt Driedger (2011).

Im Fall von Zeitrafferaudio ist es nicht notwendig, jedes einzelne akustische Ereignis festzuhalten - beim Zeitraffervideo werden auch immer einige Sekunden zwischen den Aufnahmen nicht aufgenommen. Hier besteht im Wesentlichen der Unterschied: um die extreme Zeitstauchung zu erreichen (beispielsweise 24h auf 96s, ein Faktor 900) finden sich nur einzelne Ausschnitte im Endergebnis wieder.

¹⁹<http://www.soundplusdesign.com/?p=3895>, abgerufen 18-04-23

²⁰<https://vimeo.com/13953781>, abgerufen 18-04-23

3. Rechtliche Aspekte

Auch wenn in dieser Arbeit die technischen Gesichtspunkte im Vordergrund stehen, können die rechtlichen Aspekte nicht komplett ausgeklammert werden. Das Projekt stößt dabei insbesondere auf zwei unterschiedliche Rechtsbereiche: Datenschutz und Urheberrecht. Einerseits in der zuerst behandelten Frage, welche Software verwendet werden kann, so dass eine Wiederverwendung und ein Einsatz im Unterricht möglich sind. Außerdem soll geklärt werden, ob rechtlich etwas gegen den als Beispiel gewählten Einsatz zur Aufnahme des Fortschritts einer Baustelle spricht: Die Aufnahme von Fotos zur Erstellung von Zeitrafferaufnahmen kann je nach Art des Einsatzes und der anschließenden Verwendung, insbesondere Veröffentlichung, potentiell mit verschiedenen Gesetzen in Konflikt geraten. Das kann beispielsweise eine Verletzung der Privatsphäre Dritter betreffen, die Verletzung des Urheberrechtes oder einen Verstoß gegen die Bestimmungen zur Videoüberwachung. In diesem Kapitel wird in aller Kürze auf die genannten Beispiele unter Nennung der relevanten Paragraphen eingegangen, wobei sich die Analyse auf österreichisches Recht beschränkt. Eine umfassendere Analyse der aktuellen Rechtslage zu einem verwandten Problem findet sich beispielsweise in Plasser (2017). Mit Open Educational Resources (OER) im Unterricht, insbesondere warum Lehrende offene Bildungsressourcen benötigen beschäftigt sich Grandl (2017), eine Einführung in OER gibt auch Vlaj (2014).

3.1. Urheberrecht, freie und quelloffene Software, OER Lizenzen - Urheberrechtsgesetz

OER und Free Open Source Software (FOSS) zeichnen sich zuerst dadurch aus, dass sie unter einer Lizenz veröffentlicht werden, die ohne weitere Abgeltung jedermann bestimmte Rechte einräumt. Diese müssen im Fall von FOSS entsprechend gängiger Definitionen zumindest folgende Punkte umfassen¹:

- Freiheit, das Programm auszuführen wie man möchte, für jeden Zweck (Freiheit o).

¹aus: <https://www.gnu.org/philosophy/free-sw.de.html>, abgerufen 18-04-23

3. Rechtliche Aspekte

- Freiheit, die Funktionsweise des Programms zu untersuchen und eigenen Datenverarbeitungsbefürfnissen anzupassen (Freiheit 1). Der Zugang zum Quellcode ist dafür Voraussetzung.
- Freiheit, das Programm zu redistribuieren und damit seinen Mitmenschen zu helfen (Freiheit 2).
- Freiheit, das Programm zu verbessern und diese Verbesserungen der Öffentlichkeit freizugeben, damit die gesamte Gesellschaft davon profitiert (Freiheit 3). Der Zugang zum Quellcode ist dafür Voraussetzung.

FOSS kann damit jedenfalls immer als OER gesehen werden. Der Grundgedanke ist derselbe, nur wird er bei OER genereller, also ohne programmierspezifische Aspekte gefasst (Albers u. a., 2015).

Natürlich können auch darüber hinausgehende Rechte der Allgemeinheit (oder einer Untergruppe, etwa nur für den nichtkommerziellen oder nichtmilitärischen Einsatz) gewährt werden, bis hin zur Gemeinfreiheit. Für OER gibt es mit den 5R (Retain, Reuse, Revise, Remix, Redistribute) ebenso eine (weitgehend deckungsgleiche) Liste an Rechten, die gegeben sein müssen²:

- Verwahren/Vervielfältigen – das Recht, Kopien des Inhalts anzufertigen, zu besitzen und zu kontrollieren (z. B. Download, Speicherung und Vervielfältigung)
- Verwenden – das Recht, den Inhalt in unterschiedlichen Zusammenhängen einzusetzen (z. B. im Klassenraum, in einer Lerngruppe, auf einer Website, in einem Video)
- Verarbeiten – das Recht, den Inhalt zu bearbeiten, anzupassen, zu verändern oder umzugestalten (z. B. einen Inhalt in eine andere Sprache zu übersetzen)
- Vermischen – das Recht, einen Inhalt im Original oder in einer Bearbeitung mit anderen offenen Inhalten zu verbinden und aus ihnen etwas Neues zu schaffen (z. B. beim Einbauen von Bildern und Musik in ein Video)
- Verbreiten – das Recht, Kopien eines Inhalts mit Anderen zu teilen, im Original oder in eigenen Überarbeitungen (z. B. einem Freund eine Kopie zu geben oder online zu veröffentlichen)

Die Listen sind einander sehr ähnlich: Verwahren/Vervielfältigen - Freiheit 2, Verwenden - Freiheit 0, Verarbeiten - Freiheit 1, Vermischen - Freiheit 3, Verbreiten - Freiheit 2 und Freiheit 3.

Die aus dem Bereich der Softwarelizenzen stammenden GNU-Lizenzen (etwa die GPL) haben ihren Ursprung bereits in den 1980er Jahren und nutzen das Urheberrecht auch um mit einer Klausel der/dem Lizenznehmer/in die Pflicht aufzuerlegen, jegliche

²aus <https://open-educational-resources.de/5rs-auf-deutsch/>, abgerufen 18-04-23, Original von David Wiley, in seiner „Definition of Open Content“

3. Rechtliche Aspekte

Bearbeitung des Werks unter die Lizenz des ursprünglichen Werks zu stellen - damit soll eine große Verbreitung von unter offenen Lizenzen gestellter Software erreicht werden.

Heute gängiger sind sowohl für Software, OER, sowie jedwede andere Inhalte, die jemand unter einer entsprechenden Lizenz zur Verfügung stellen möchte, die Creative-Commons-Lizenzen (kurz CC). Online kann auf den Seiten der gleichnamigen NGO eine Variante ausgewählt werden, die den jeweiligen Absichten entspricht.

Für alle Teile dieser Arbeit, deren Urheberrecht beim Autor liegen, ist das die Variante "Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Österreich (CC BY-SA 3.0 AT)"³. Damit kann das beschriebene Projekt nachgebaut, die Software und Grafiken oder Fotos verwendet, adaptiert, und wieder veröffentlicht werden. Zu beachten ist lediglich die Namensnennung, und dass Änderungen, sofern sie veröffentlicht werden, ebenso unter die genannten Lizenzbedingungen gestellt werden müssen. Das gilt also beispielsweise für Abbildung 7.1, nicht aber für Abbildungen 2.1 und 2.2, da für diese das Urheberrecht nicht beim Autor liegt (sie sind ein Bildzitat gemäß §42f Urheberrechtsgesetz). Da bei Abbildung 2.1 aber ohnehin zumindest jedenfalls nach österreichischem Recht aufgrund des bereits vor 114 Jahren verstorbenen Urhebers keine Urheberrechtsansprüche mehr geltend gemacht werden können, besteht hier nicht einmal die lizenzrechtliche Notwendigkeit der Namensnennung oder der Weitergabe unter gleichen Bedingungen, denn das Urheberrechtsgesetz §60 bestimmt: "Das Urheberrecht an Werken der Literatur, der Tonkunst und der bildenden Künste endet siebenzig Jahre nach dem Tod des Urhebers (§10 Abs. 1). Bei einem von mehreren Urhebern gemeinsam geschaffenen Werk (§11) endet das Urheberrecht siebenzig Jahre nach dem Tod des letztlebenden Miturhebers (§10 Abs. 1)."⁴

3.2. Persönlichkeitsrecht und Privatsphäre - Allgemeines bürgerliches Gesetzbuch, Europäische Menschenrechtskonvention und Mediengesetz

Ganz grundlegend schützt §16 ABGB das allgemeine Persönlichkeitsrecht als das angeborene Recht auf Freiheit und ein nicht fremdbestimmtes Leben: „Jeder Mensch hat angeborne, schon durch die Vernunft einleuchtende Rechte, und ist daher als eine

³Ausgenommen sind nur die Abbildungen 2.1, 2.2, 8.2 und 8.13

⁴RIS: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10001848>, abgerufen 18-04-23

3. Rechtliche Aspekte

Person zu betrachten. Slavery oder Leibeigenschaft, und die Ausübung einer darauf sich beziehenden Macht, wird in diesen Ländern nicht gestattet.“⁵

Gleichfalls im ABGB ist das Recht auf Wahrung der Privatsphäre in §1328a normiert: „Wer rechtswidrig und schuldhaft in die Privatsphäre eines Menschen eingreift oder Umstände aus der Privatsphäre eines Menschen offenbart oder verwertet, hat ihm den dadurch entstandenen Schaden zu ersetzen. Bei erheblichen Verletzungen der Privatsphäre, etwa wenn Umstände daraus in einer Weise verwertet werden, die geeignet ist, den Menschen in der Öffentlichkeit bloßzustellen, umfasst der Ersatzanspruch auch eine Entschädigung für die erlittene persönliche Beeinträchtigung.“⁶

Hier ist natürlich die Frage nach der Schuldhaftigkeit zentral - soll eine Aufnahme beispielsweise einen Baufortschritt dokumentieren, filmt aber auch Personen auf eine ihre Privatsphäre verletzende Art, so liegt zumindest keine Absicht vor - eine Fahrlässigkeit könnte aber dennoch als schuldhaft ausgelegt werden. Artikel 8 Abs. 1 der Europäischen Menschenrechtskonvention (EMRK) garantiert das Recht auf Achtung des Privat- und Familienlebens: „Jedermann hat Anspruch auf Achtung seines Privat- und Familienlebens, seiner Wohnung und seines Briefverkehrs.“⁷

In eine ähnliche Richtung des Schutzes der Privatsphäre wird auch im §7 Mediengesetz der Schutz des höchstpersönlichen Lebensbereiches normiert: „Wird in einem Medium der höchstpersönliche Lebensbereich eines Menschen in einer Weise erörtert oder dargestellt, die geeignet ist, ihn in der Öffentlichkeit bloßzustellen, so hat der Betroffene [sic!] gegen den Medieninhaber [sic!] Anspruch auf eine Entschädigung für die erlittene Kränkung.“⁸

Das Mediengesetz bezieht sich dabei natürlich auf den Aspekt der Veröffentlichung und verbietet auch nur Aufnahmen, die nicht nur veröffentlicht werden, sondern auch geeignet sind, jemanden in der Öffentlichkeit bloßzustellen.

⁵RIS: <https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Bundesnormen&Dokumentnummer=NOR12017706>, abgerufen 18-04-23

⁶RIS: <https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Bundesnormen&Dokumentnummer=NOR40045309>, abgerufen 18-04-23

⁷RIS: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10000308>, abgerufen 18-04-23

⁸RIS: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10000719>, abgerufen 18-04-23

3.3. Recht am eigenen Bild und Urheberrecht - Urheberrechtsgesetz

Das Problem, ungewollt auf einem Foto abgebildet zu werden, ist nicht neu. Die große Verbreitung von digitalen Kameras, insbesondere auch in Mobiltelefonen, und die geringen Kosten für einzelne Fotos führen aber zu einer ganz anderen Häufigkeit an möglichen Verletzungen des Rechtes am eigenen Bild: In §78 Urheberrechtsgesetz (UrhG) wird dieses eingeführt indem die Verbreitung von Fotos von Personen, wenn dadurch deren berechnigte Interessen verletzt werden, untersagt wird: "Bildnisse von Personen dürfen weder öffentlich ausgestellt noch auf eine andere Art, wodurch sie der Öffentlichkeit zugänglich gemacht werden, verbreitet werden, wenn dadurch berechnigte Interessen des Abgebildeten oder, falls er gestorben ist, ohne die Veröffentlichung gestattet oder angeordnet zu haben, eines nahen Angehörigen verletzt würden."⁹

Wiederum, ähnlich wie im Mediengesetz, ist nicht die Aufnahme untersagt, sondern nur die öffentliche Ausstellung und zwar auch nur dann, wenn berechnigte Interessen der fotografierten Person dagegensprechen. Das Recht am eigenen Bild, wie es an dieser Stelle im Gesetz steht, ist aber eigentlich keine Frage des Urheberrechts. Auch dieses kann aber Thema sein, nämlich dann, wenn urheberrechtlich geschützte Werke fotografiert (und veröffentlicht) werden. Im Beispiel von Zeitrafferaufnahmen von (entstehenden) Bauten sieht das Urheberrechtsgesetz aber mit der Panoramafreiheit in §54 Abs. 1 Z. 5 UrhG eine Ausnahme vor: "Werke der Baukunst nach einem ausgeführten Bau oder andere Werke der bildenden Künste nach Werkstücken, die dazu angefertigt wurden, sich bleibend an einem öffentlichen Ort zu befinden, zu vervielfältigen, zu verbreiten, durch optische Einrichtungen öffentlich vorzuführen, durch Rundfunk zu senden und der Öffentlichkeit zur Verfügung zu stellen; ausgenommen sind das Nachbauen von Werken der Baukunst, die Vervielfältigung eines Werkes der Malkunst oder der graphischen Künste zur bleibenden Anbringung an einem Orte der genannten Art sowie die Vervielfältigung von Werken der Plastik durch die Plastik."¹⁰

3.4. Bildaufnahme und Videoüberwachung - Datenschutzgesetz und Datenschutz-Anpassungsgesetz

Im Datenschutzgesetz 2000 wird Videoüberwachung in §50a auf eine Art definiert, unter die die Zeitrafferaufnahme jedenfalls technisch fällt: "Videoüberwachung im

⁹RIS: <https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Bundesnormen&Dokumentnummer=NOR12024485>, abgerufen 18-04-23

¹⁰RIS: <https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Bundesnormen&Dokumentnummer=NOR40173320>, abgerufen 18-04-23

3. Rechtliche Aspekte

Sinne dieses Abschnittes bezeichnet die systematische, insbesondere fortlaufende Feststellung von Ereignissen, die ein bestimmtes Objekt (überwachtes Objekt) oder eine bestimmte Person (überwachte Person) betreffen, durch technische Bildaufnahme- oder Bildübertragungsgeräte. Für derartige Überwachungen gelten die folgenden Absätze, sofern nicht durch andere Gesetze Besonderes bestimmt ist.“¹¹ Dennoch wird beispielsweise die Dokumentation des Baufortschritts kaum als Videoüberwachung – die ja eine ganz andere Intention verfolgt – zu interpretieren sein, insbesondere wenn sie nicht gewerblich erfolgt - eine wichtige Ausnahme dahingehend räumt §45 DSGVO ein: “Für ausschließlich persönliche oder familiäre Tätigkeiten dürfen natürliche Personen Daten verarbeiten, wenn sie ihnen vom Betroffenen [sic!] selbst mitgeteilt wurden oder ihnen sonst rechtmäßigerweise, insbesondere in Übereinstimmung mit §7 Abs. 2, zugekommen sind.“¹²

Im Mai 2018 werden die Bestimmungen des DSGVO 2000 allerdings obsolet. Von nun an gibt EU-weit die Datenschutzgrundverordnung (DSGVO) die rechtlichen Rahmenbedingungen vor, in Österreich das Datenschutz-Anpassungsgesetz 2018¹³. Es spricht jetzt nicht mehr von Videoüberwachung, sondern allgemeiner von Bildverarbeitung und regelt in §30 wann die eine Bildaufnahme überhaupt zulässig ist, nämlich beispielsweise, wenn

- „Im Einzelfall überwiegende berechtigte Interessen des Verantwortlichen oder eines Dritten bestehen und die Verhältnismäßigkeit gegeben ist.“
- „Sie ein privates Dokumentationsinteresse verfolgt, das nicht auf die identifizierende Erfassung unbeteiligter Personen oder die gezielte Erfassung von Objekten, die sich zur mittelbaren Identifizierung solcher Personen eignen, gerichtet ist.“

Hingegen unzulässig ist gemäß §30 Abs. 4:

- „eine Bildaufnahme ohne ausdrückliche Einwilligung der betroffenen Person in deren höchstpersönlichen [sic!] Lebensbereich,
- eine Bildaufnahme zum Zweck der Kontrolle von Arbeitnehmern,
- der automationsunterstützte Abgleich von mittels Bildaufnahmen gewonnenen personenbezogenen Daten mit anderen personenbezogenen Daten oder
- die Auswertung von mittels Bildaufnahmen gewonnenen personenbezogenen Daten anhand von besonderen Kategorien personenbezogener Daten (Art. 9 DSGVO) als Auswahlkriterium.“

¹¹RIS: <https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Bundesnormen&Dokumentnummer=NOR40113740>, abgerufen 18-04-23

¹²RIS: <https://www.ris.bka.gv.at/NormDokument.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10001597&FassungVom=2018-02-05&Artikel=2&Paragraf=45&Anlage=&Uebergangsrecht=>, abgerufen 18-04-23

¹³RIS: https://www.ris.bka.gv.at/Dokumente/RegV/REGV_C00_2026_100_2_1367515/REGV_C00_2026_100_2_1367515.html, abgerufen 18-04-23

3. Rechtliche Aspekte

Das in dieser Arbeit dokumentierte Beispiel sollte gemäß §30 Abs 3 lit. 3 also jedenfalls im Sinne der neuen Rechtslage zulässig sein. Eine Unzulässigkeit könnte sich aber gemäß Abs. 4 lit. 2 aus der Kontrolle von Arbeitnehmer/innen ergeben - das stellt aber keinesfalls den Zweck der Aufnahme dar.

3.5. Strafrecht - Strafgesetzbuch

Im Strafgesetzbuch ist in §120 für unerlaubte Tonaufnahme eine Freiheitsstrafe von bis zu einem Jahr vorgesehen: "Wer ein Tonaufnahmegerät oder ein Abhörgerät benützt, um sich oder einem anderen Unbefugten von einer nicht öffentlichen und nicht zu seiner Kenntnisnahme bestimmten Äußerung eines anderen Kenntnis zu verschaffen, ist mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bis zu 720 Tagessätzen zu bestrafen."¹⁴

Eine vergleichbaren strafrechtlichen Schutz gegen auf ähnliche Art angefertigte Bildaufnahmen gibt es allerdings nicht. Damit ist die bloße Herstellung von Fotos bisher jedenfalls nicht strafbar.¹⁵

3.6. Rechtsprechung

Im Jahr 2013 hat der Oberste Gerichtshof festgestellt, dass zwar prinzipiell die bloße Aufnahme einer Person - im Gegensatz zur Verbreitung der Aufnahme - zulässig ist, da §78 UrhG keinen Schutz gegen die ungewollte Aufnahme von Bildern gibt, sehr wohl aber basierend auf §16 ABGB und Artikel 8 EMRK „bereits die Herstellung eines Bildnisses ohne Einwilligung des Abgebildeten [sic!] einen unzulässigen Eingriff in dessen allgemeines Persönlichkeitsrecht darstellen [kann]. Dabei wird das allgemeine Persönlichkeitsrecht des Betroffenen [sic!] nicht nur dann verletzt, wenn Abbildungen einer Person in deren privatem Bereich angefertigt werden, um diese der Öffentlichkeit zugänglich zu machen. Vielmehr kann auch die Herstellung von Bildnissen einer Person in der Öffentlichkeit zugänglichen Bereichen und ohne Verbreitungsabsicht einen unzulässigen Eingriff in das Persönlichkeitsrecht des Betroffenen [sic!] darstellen. Schon das damit verbundene fotografische Festhalten einer bestimmten Tätigkeit oder Situation kann vom Abgebildeten als unangenehm empfunden werden und ihn an der freien Entfaltung seiner Persönlichkeit hindern."¹⁶ Weiters stellt der OGH aber fest, dass

¹⁴RIS: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10002296>, abgerufen 18-04-23

¹⁵vgl.: <https://rdb.manz.at/document/rdb.tso.LIoejz20170504>, abgerufen 18-04-23

¹⁶aus https://rdb.manz.at/document/ris.just.JJT_20130227_OGH0002_00600B00256_12H0000_000, abgerufen 18-04-23

3. Rechtliche Aspekte

es einer Abwägung im Einzelfall bedarf: „Dabei bedarf es allerdings - wie stets bei der Ermittlung von Umfang und Grenzen von Persönlichkeitsrechten - einer umfassenden Güter- und Interessenabwägung im Einzelfall. Hierbei kommt es zunächst darauf an, ob der Abgebildete [sic!] auf der Aufnahme zu identifizieren ist. Je weniger deutlich dies der Fall ist, umso geringer ist die Beeinträchtigung. Außerdem ist zu berücksichtigen, ob die Aufnahme gezielt erfolgt oder eine Person nur zufällig auf ein Bild gerät. Im ersteren Fall wird ein Gefühl der Überwachung vermittelt, das den Abgebildeten [sic!] an der freien Entfaltung seiner Persönlichkeit hindert. In diesem Sinne hat auch der Bundesgerichtshof darauf abgestellt, dass dann, wenn vorübergehende Passanten [sic!] zufällig in eine Aufnahme miteinbezogen werden, diese ohne weiteres hinnehmen müssen, wenn sie öffentlichen Wegeraum benützen. Ist der Abgebildete [sic!] überhaupt nicht mehr zu identifizieren - wie etwa bei Urlaubsfotos Außenstehende [sic!] Personen im Hintergrund der Aufnahme - scheidet eine Persönlichkeitsrechtsverletzung in der Regel jedenfalls dann aus, wenn der Abgebildete [sic!] nicht den Eindruck erhält, er [sic!] werde gezielt fotografiert.“¹⁷

Im Fall der Zeitrafferaufnahme eines Baufortschritts ist, zumindest mit dem in dieser Arbeit vorgestellten Projekt, auszuschließen, dass Abgebildete deutlich zu erkennen sind. Ihre Aufnahme erfolgt außerdem nicht gezielt, es entsteht kein Gefühl der Überwachung. Vielmehr werden neben Bauarbeiter/innen vorübergehende Passant/innen zufällig aufgenommen – es entsteht nicht der Eindruck einer gezielten Fotografie.

Ein weiteres höchstgerichtliches Urteil aus dem Jahr 1997 bezieht sich auf geheime Bildaufnahmen im Privatbereich und ist damit für den als Beispiel gewählten Einsatzzweck nicht relevant.¹⁸

Zur systematischen, verdeckten, identifizierenden Videoüberwachung stellt der OGH fest, dass damit immer ein Eingriff in das geschützte Recht auf Achtung der Geheimosphäre gegeben ist. „Die Videoaufzeichnung ist identifizierend, wenn sie auf Grund eines oder mehrere Merkmale letztlich einer bestimmten Person zugeordnet werden kann“¹⁹. Auch diese Voraussetzung – verdeckt und identifizierend - ist im vorliegenden Beispiel nicht gegeben.

Von der Datenschutzkommission (DSK, jetzt: Datenschutzbehörde) wurde 2012 der Beschluss gefasst, eine zur Videoüberwachung des Umfeldes eines Autos angebrachte

¹⁷aus https://rdb.manz.at/document/ris.just.JJT_20130227_OGH0002_00600B00256_12H0000_000, abgerufen 18-04-23

¹⁸RIS: https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Justiz&Dokumentnummer=JJR_19970130_OGH0002_00600B02401_96Y0000_001, abgerufen 18-04-23

¹⁹RIS: https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Justiz&Dokumentnummer=JJR_20051219_OGH0002_00800B00108_05Y0000_001, abgerufen 18-04-23

3. Rechtliche Aspekte

sogenannte Dashcam als Datenanwendung abzulehnen²⁰. Anders als in diesem Fall, wo der Antragsteller sogar unter Hinweis auf das DSGVO die Dashcam als Videoüberwachung anzumelden versucht hat, liegt hier eine andere Absicht vor. Wenn auch technisch ähnlich, so ist der Zweck des in dieser Arbeit dokumentierten Beispiels aber in keiner Weise die Überwachung, Verhinderung oder Aufklärung von Straftaten.

3.7. Zusammenfassung

Das Beispiel - die Zeitrafferaufnahme einer Baustelle mit dem in dieser Arbeit beschriebenen Gerät - verstößt wahrscheinlich gegen keine bestehenden Gesetze. Am ehesten könnte ein Gericht in seiner Abwägung Persönlichkeitsrechte oder Datenschutzrecht verletzt sehen. Der Autor geht davon aus, gegen kein Gesetz verstoßen zu haben - und selbst wenn, so gilt wahrscheinlich und beendet zugleich den rechtlichen Exkurs standesgemäß: *Nullus in causa, nullus iudex*.²¹

²⁰RIS: https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Dsk&Dokumentnummer=DSKTE_20121107_K600319_005_0002_DVR_2012_00, abgerufen 18-04-23

²¹Wo kein Kläger, da kein Richter.

4. Wahl der Hard- und Software

In diesem Kapitel werden verschiedene Optionen für die Kernkomponenten der verwendeten Hard- und Software, sowie Audio- und Videoformate diskutiert. Da freie Software eingesetzt werden soll, bestimmen alleine die Hardwarekomponenten den Preis. Um das Projekt für den Unterricht in einem vertretbaren Rahmen zu halten, sollen die Gesamtkosten bei nicht mehr als etwa 50 Euro liegen¹ und der Nachbau zeitlich in einem ähnlichen Rahmen bleiben, wie typische Projekte im Werkunterricht². Die vielfältigen Erweiterungsmöglichkeiten - einige Beispiele nennt der zweite Teil dieser Arbeit - sollen zulassen, das Thema leicht auf den Umfang einer vorwissenschaftlichen Arbeit (VWA) auszuweiten.

4.1. Hardware

In diesem Kapitel werden die wichtigsten verwendeten Hardwarekomponenten beschrieben. Dazu werden jeweils die Anforderungen und mögliche Alternativen genannt, sowie abschließend die erfolgte Wahl begründet. Über alle Komponenten hinweg sind Preis, Verfügbarkeit und Kompatibilität zum RPI relevante Aspekte.

4.1.1. Computer

Das Ziel der Arbeit setzt bereits den RPI als Plattform fest, in der Einleitung und im Kapitel zur Geschichte finden sich auch Hinweise, warum es sich dabei um eine sehr geeignete Wahl handelt - mit dem RPI gibt es einen kostengünstigen, breit verfügbaren Computer mit einer großen und aktiven Gemeinschaft die ihn insbesondere auch im Bereich der Informatikausbildung einsetzt. Ganz besonders sticht dabei aber auch die Qualität des RPI als Prototypenplattform heraus: "The Raspberry Pi is an ideal platform for this as it is commodity hardware, supports high-level programming languages (e.g.,

¹Das entspricht laut <https://www.bmfj.gv.at/familie/finanzielle-unterstuetzungen/schulbuchaktion.html> (abgerufen 18-04-24) den minimale Kosten für Schulbücher pro Schüler/in und Jahr, erscheint also als ein vertretbarer finanzieller Aufwand.

²Wie beispielsweise den unter https://www.schule.at/fileadmin/DAM/Gegenstandsportale/Gender_und_Bildung/Dateien/Werken-Konkret.internet.pdf (abgerufen 18-04-24) genannten

4. Wahl der Hard- und Software

Python) and runs popular variants of Unix-like operating systems.” (Johnston und Cox, 2017)

Technisch betrachtet stellt - abgesehen von gebrauchter Hardware - der RPI Zero mit einem Preis von nur etwa 5 USD eine der aktuell kostengünstigsten verfügbaren Optionen dar, um ein Projekt wie dieses umzusetzen. Aktuelle vergleichbare Einplatinencomputer sind von verschiedenen Herstellern und in diversen Varianten verfügbar, allerdings verfügt keine dieser Plattformen über das selbe Angebot an Zubehör und bei weitem nicht über eine vergleichbare Community. Einige Beispiele:

- **CHIP** ist ein Einplatinencomputer, der zu Beginn zu einem Preis von 9 USD erhältlich war, aktuell aber nicht mehr verfügbar ist.
- In direkter Konkurrenz zum RPI Zero W steht - darauf deutet auch der Name hin - der **Orange Pi Zero**³, lieferbar aus China inkl. Versand um etwa 15 Euro. Er verfügt in dieser Variante auch über nur 512 MB Ram, hat aber einen 4-Kern Prozessor, Ethernet mit Power over Ethernet (POE), dafür aber keinen ausgeführten Videoausgang.
- Das Modell M2+ EDU von **Banana Pi** ist ab etwa 25 Euro erhältlich⁴, verfügt wie der RPI Zero über nur 512 MB Ram, dafür aber über 4 CPU Kerne.
- Das **Tinker Board** von ASUS ist deutlich leistungsfähiger (1.8 GHz CPU mit 4 Kernen, 2 GB Arbeitsspeicher) und mit einem Preis von knapp 60 Euro⁵ auch deutlich teurer als der RPI Zero.

Die Wahl für dieses Projekt fällt auf den RPI Zero⁶. Für die etwas anspruchsvollen weiterführenden Ideen kommt ein RPI 2 zum Einsatz der über viermal so viele CPU-Kerne und doppelt so viel Arbeitsspeicher verfügt.

4.1.2. Kamera

Zentral für die spätere Qualität der Aufnahmen ist natürlich die Kamera. Zu einem Preis von etwa 30 USD bietet die vom Hersteller des RPI entwickelte Kamera⁷ einen 8 MP, 1/4" Sony IMX219 image sensor. Außerdem gibt es zahlreiche günstigere Nachbauten, die meistens auf dem etwas älteren Modell mit 5 MP basieren, einige davon werden mit einer Weitwinkellinse angeboten.

³<http://www.orangepi.org/orangepizero/>, abgerufen 18-04-24

⁴etwa hier https://www.reichelt.at/?ARTICLE=191585&PROVID=2774&utm_source=psuma&utm_medium=Geizhals.at, abgerufen 18-04-24

⁵etwa hier erhältlich https://www.reichelt.at/?ARTICLE=194619&PROVID=2774&utm_source=psuma&utm_medium=Geizhals.at, abgerufen 18-04-24

⁶beziehungsweise auf die Variante RPI Zero W, da die Verfügbarkeit im Wlan die Arbeit mit dem Gerät wesentlich vereinfacht - alternativ müssten oft eine Tastatur und ein Monitor angeschlossen werden

⁷erhältlich z. B. bei <https://www.adafruit.com/product/3099>, abgerufen 18-04-24

4. Wahl der Hard- und Software

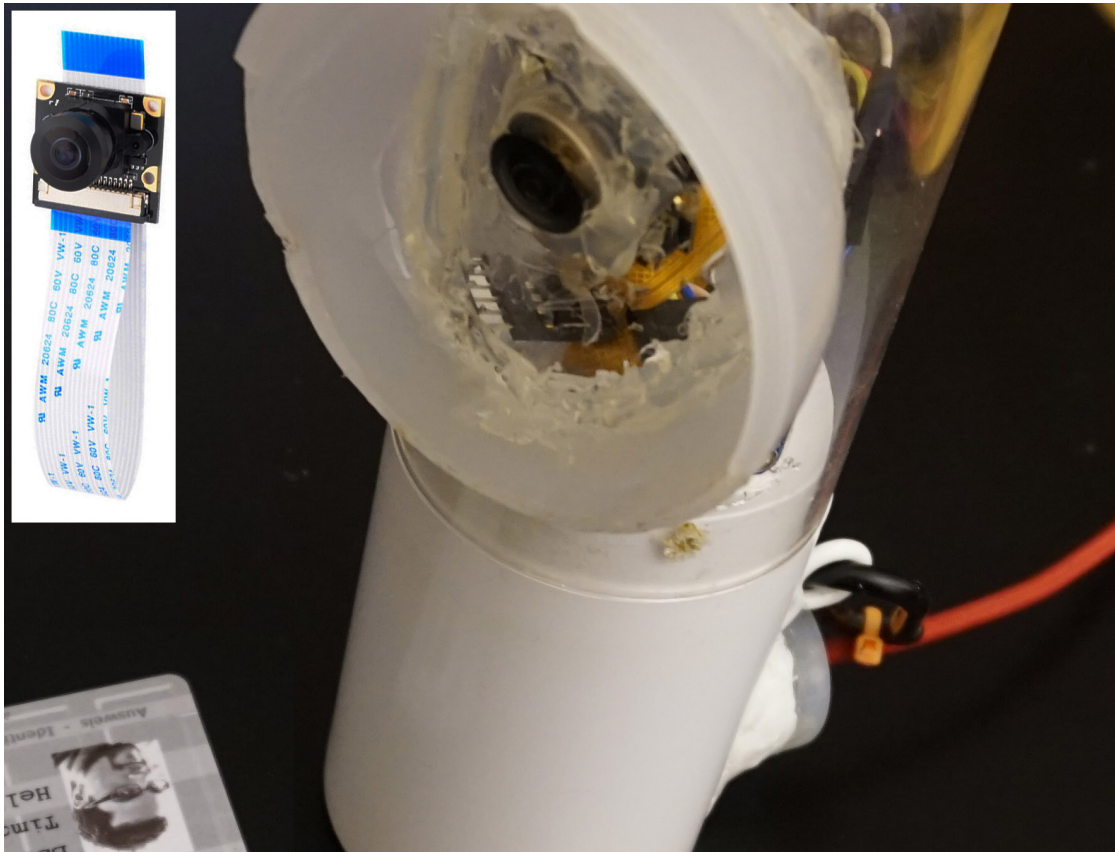


Abbildung 4.1.: RPI Kamera

Eine ungleich höherqualitative Alternative stellen Digitalkameras dar. Für Geräte von Canon existiert mit CHDK eine alternative Firmware, die es erlaubt, die Kamera mittels RPI zu steuern und Fotos via USB auf den RPI zu überspielen. Damit können die Aufnahmen auch mit einer solchen Kamera erfolgen - erfolgreich getestet wurde das mit dem Modell Canon PowerShot SX230 HS. Ein Nachteil ist der Preis - leider müsste man auf gebrauchte Geräte zurückgreifen, da die günstigsten neuen Geräte knapp 100 Euro kosten. Außerdem würden Stromversorgung und die Größe die Möglichkeiten insgesamt stark einschränken.

Die Wahl fällt daher auf einen Nachbau der RPI Kamera mit 5 Megapixeln, dargestellt in Abbildung 4.1 der über ein 160 Grad Weitwinkelobjektiv verfügt: Damit ist es möglich, die ganze ausgewählte Baustelle als Testszene aufzunehmen.

4.1.3. Unterbrechungsfreie Stromversorgung (USV)

Zur USV von Geekworm gibt es im selben Preissegment von etwa 10 Euro⁸ keine vergleichbaren Konkurrenzprodukte. Mit einem separat erstandenen LiPo Akku kann der RPI Zero für mindestens 10 Stunden getrennt vom Netz normal weiterarbeiten. Da der Batteriestand ausgelesen werden kann, ist es auch möglich, ein automatisches Herunterfahren ab einem kritischen Ladungsstand einzuleiten.

Dennoch bestehen andere, erstaunlich einfache Möglichkeiten, um zumindest kurze Unterbrechungen der Stromversorgung zu überbrücken: Da ein RPI bereits ab einer Spannung von 4.6V (getestet mit einem Voltmeter, entspricht auch den Herstellerangaben) keine Unterspannung mehr meldet und zumindest bis zu den an vielen USB-Netzteilen noch üblichen 5.25V Spannung einwandfrei arbeitet, kann die Differenz ausgenutzt werden, um mittels Kondensatoren in der Größenordnung von etwa 100 F bis zu etwa einer Minute⁹ die unterbrochene Stromversorgung abgefangen werden. Abbildung 4.2 stellt den simplen Aufbau mit zwei 100 F Kondensatoren in Serie (um eine Spannungsfestigkeit von mehr als 5V zu erreichen) dar. Entsprechende Kondensatoren werden zunehmend erschwinglich, mit Preisen im einstelligen Eurobereich. Es ist zwar kaum möglich, eine noch simplere Lösung zu finden, da alle Nachteile von Akkus und Batterien (Ladezyklen etc.) entfallen. Das Beispiel hat auch didaktischen Wert, da Kondensatoren ein grundlegendes elektrotechnisches Element sind. Dennoch überwiegen die Nachteile aber: Gefahr von extrem hohen Strömen im Kurzschlussfall, sehr hohe Ladeströme, schlechte Ausnutzung der vorhandenen Kapazität (nur im Bereich von 4.6 bis 5.25 V, die von 4.6 bis 0 V noch vorhandene Ladung wird nicht entnommen.)

Daher fällt die Wahl auf die USV von Geekworm, sie ist in Abbildung 4.3 (Aufschrift UPS HAT) zu erkennen.

4.1.4. Gehäuse

Geeignet sind jede Art von wasserdichtem und nach Möglichkeit aus einfach zu bearbeitendem Material hergestellte Gefäße: Aufbewahrungsdosen, Verpackungen (ganz im Sinne des Upcycling), Trinkflaschen. Je nach Anforderungen kann auch ganz auf ein (fast) geschlossenes Gehäuse verzichtet werden. Neben dem Schutz vor Wasser und Staub oder Tieren ist auch die Temperaturentwicklung zu beachten. Ein ganz geschlossenes Gehäuse kann infrage kommen, wenn beispielsweise im Sommer Beschattung

⁸www.banggood.com/Geekworm-UPS-HAT-Board-For-Raspberry-Pi-3-Model-B-Pi-2B-B-A-p-1105264.html, abgerufen 18-04-24

⁹getestet an einem RPI 2 - dementsprechend sind mit einem RPI Zero deutlich längere Zeiträume zu erwarten, da die Leistungsaufnahme entsprechend geringer ist

4. Wahl der Hard- und Software

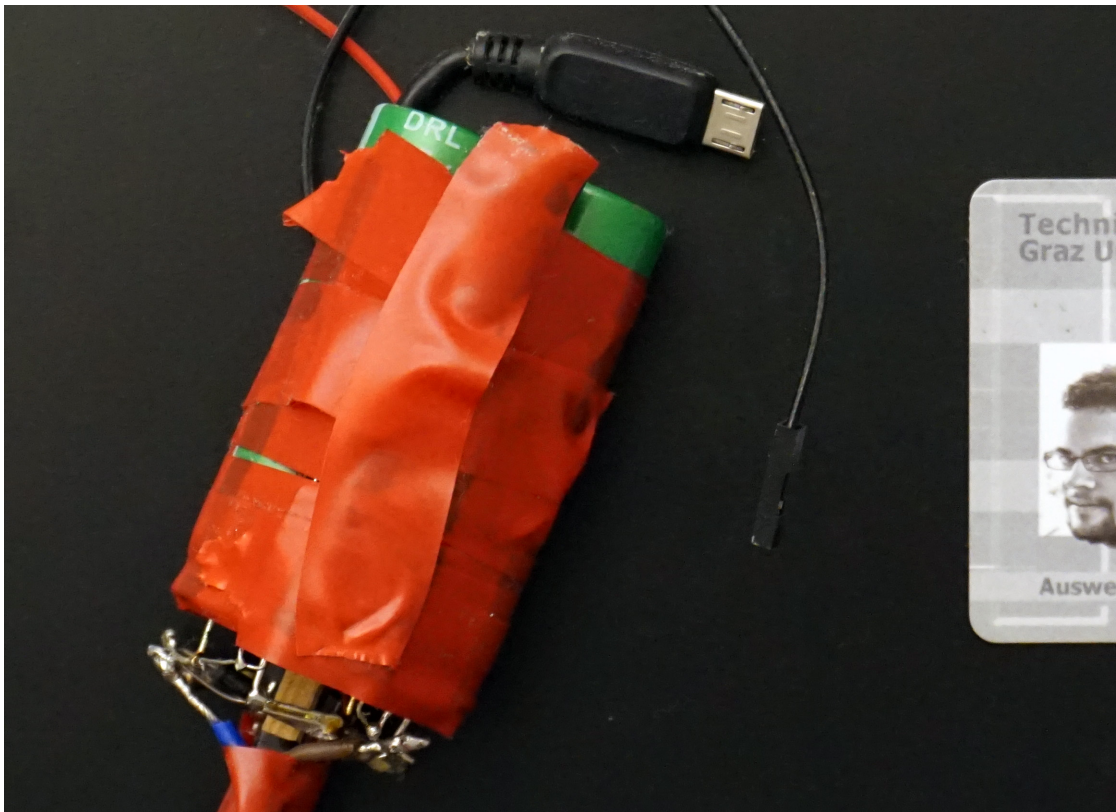


Abbildung 4.2.: Kondensatoren zur RPI Energieversorgung

4. Wahl der Hard- und Software

sichergestellt ist. Auch die Farbe und das Material selbst haben natürlich einen Einfluss auf die Temperaturentwicklung. Ein nicht verfolgter Ansatz wäre ein bewusst groß ausgelegtes aber geschlossenes Gehäuse, möglichst mit einem Teil der Oberfläche in gut wärmeleitfähigem Metall ausgeführt. Dann kann gehäuseintern ein kleiner Lüfter für ausreichend Zirkulation sorgen, um alle Bauteile über die große und (teilweise) gut thermisch leitende Gehäuseoberfläche zu kühlen. Neben der Größe, dem zusätzlichen Energieverbrauch durch den Lüfter und der Komplexität (idealerweise ist auch eine temperaturgesteuerte Regelung vorzusehen) könnten auch die Geräuschemissionen ein Problem darstellen, wenn die Mikrofone nicht ausreichend entkoppelt sind.

Gewählt wurde schließlich die Verpackung einer inzwischen nicht mehr produzierten "HTC RE"¹⁰, einer kleinen Kamera aus der populären Sparte der Action Cameras¹¹. Das Gehäuse, dargestellt in Abbildung 4.3 schließt spritzwassergeschützt. Lediglich über die Öffnung für den Stromanschluss könnte im Fall von extrem starkem Regen Wasser eindringen.

4.1.5. Akku

Bei der Wahl des Akkus sind Gewicht, Größe, Kapazität, Temperaturverhalten, Preis, elektrische Eigenschaften (Leistung, Ladestrom) und chemische Eigenschaften (Lebensdauer in Ladezyklen, notwendige Ladeelektronik) zu berücksichtigen. Die Wahl der USV gibt den Batterietyp (Lithium, Nennspannung von 3.7 V) bereits vor. Gewicht und Größe müssen mit dem gewünschten Gehäuse abgestimmt sein und geben eine Grenze für die Kapazität vor. Es ist eine Leistungsaufnahme von bis zu 2 W, also maximal 0.6 A, zu erwarten und eine übliche Leistung von deutlich unter 1 W. Die Wahl fällt daher auf einen Lithiumakku des Typs 18650 (beschreibt die Abmessungen in mm: 18,6 x 65,2)¹² - einer aktuell sehr verbreiteten Bauform. Das Modell Panasonic NCR18650B 3350mAh - 6.7A für etwa 4 Euro gehört zu den Varianten mit sehr hoher Kapazität (12.4 WH) und dafür verhältnismäßig niedriger maximaler Leistungsabgabe: Diese liegt mit 6.7 A, also einer Leistung von knapp 25W ohnehin fast um einen Faktor 10 über den Anforderungen. Dieser Akku kann laut Datenblatt bei Temperaturen von -20 bis 60 Grad eingesetzt werden, laden ist in einem engeren Temperaturfenster von 0 bis 40 Grad möglich. Das Gerät sollte mit diesem Akku für etwa 10 Stunden versorgt werden können. Der verbaute Akku ist in Abbildung 4.4 dargestellt.

¹⁰Herstellerwebsite <http://www.htc.com/us/re/re-camera/>, abgerufen 18-4-24

¹¹Ein reizvoller Gedanke auch, da es sich bei dieser Kamera um ein Gerät mit teilweise gleichen Funktionen (Zeitrafferaufnahmen) handelt, es aber um einen wesentlich beschränkten Funktionsumfang verfügt und mehr kostet.

¹²Datenblatt: <https://www.mouser.at/datasheet/2/218/-746171.pdf>, abgerufen 18-04-24

4. Wahl der Hard- und Software



Abbildung 4.3.: Gehäuse

4. Wahl der Hard- und Software

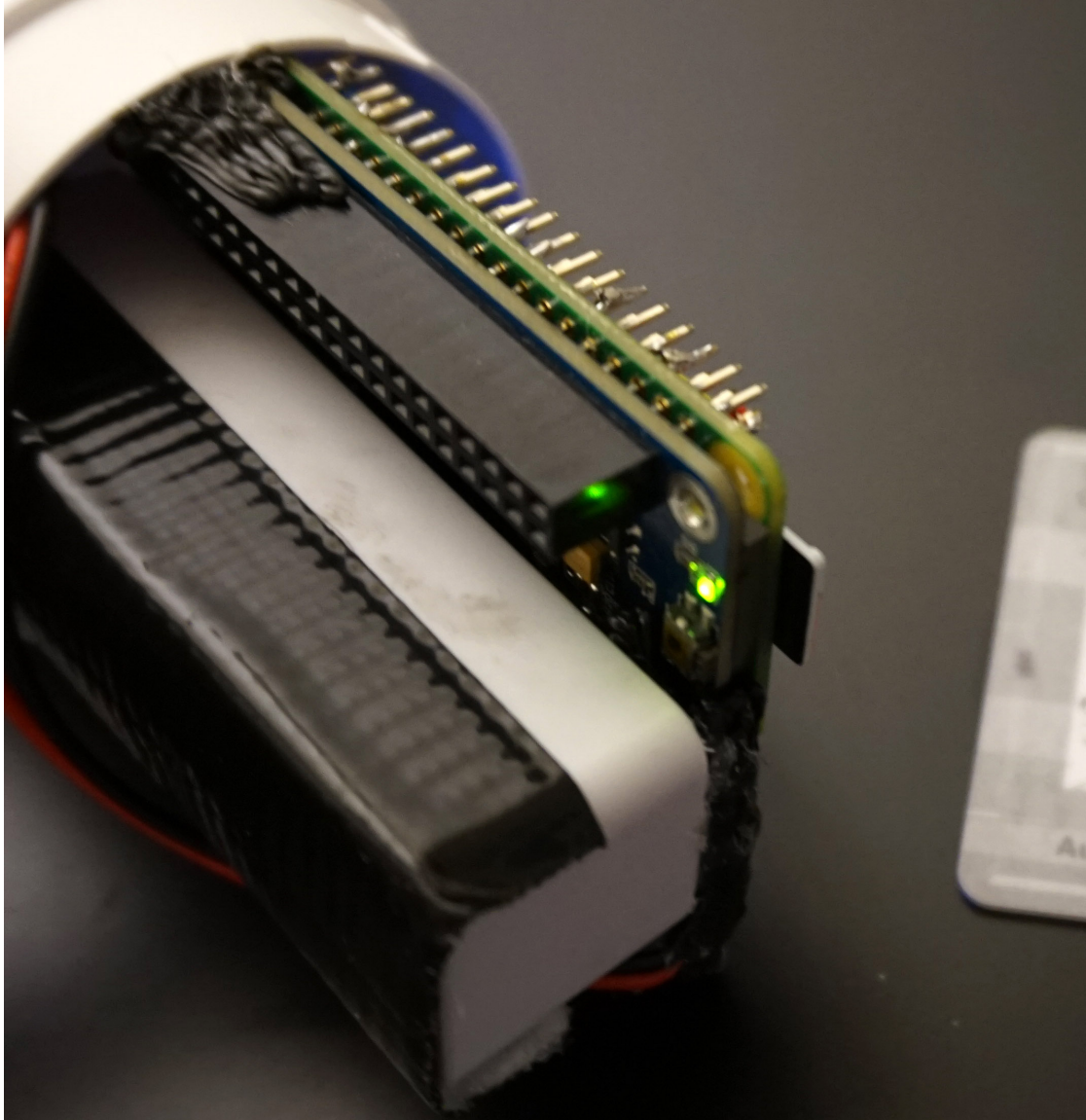


Abbildung 4.4.: RPI LiPo Akku

4.1.6. Mikrofon

Bei der Wahl des Mikrofons schränkt der Preis die komplexeren Fragen etwa nach der Richtcharakteristik stark ein. Da der RPI Zero über keinen analogen Mikrofoneingang einer Soundkarte verfügt, wäre um ein Mikrofon direkt anzuschließen eine Lösung über die Analogen GPIO denkbar, aber sicher nicht von zufriedenstellender Tonqualität. Daher ist ein Mikrofon mit digitaler Schnittstelle zu wählen. Infrage kommen dabei vor allem die zahlreich verfügbaren und sehr kostengünstigen USB-Mikrofone. Da am RPI Zero nur ein als Micro-USB ausgeführter USB-Anschluss verfügbar ist (ein zweiter dient lediglich der Stromversorgung des RPI) wäre aber zusätzlich jedenfalls ein Adapter oder ein USB-Hub zu verwenden.

Der RPI unterstützt allerdings auch die direkte Verbindung mit digitalen Mikrofonen, die über den I2S Standard kommunizieren. Zwei Mikrofone dieser Art wurden - um Stereoaufnahmen machen zu können - schließlich ausgewählt: Das winzige Mikrofon vom Typ SPHo645LM4H kann bereits für knapp 2 Euro erstanden werden, in einer Ausführung die sich leichter, das heißt ohne Reflow-Ofen, anschließen lässt, steigen die Kosten auf etwa 5 Euro.

Das gewählte Mikrofon zeichnet sich durch die sehr kleinen Abmessungen von nur 3.50 x 2.65 x 0.98 mm und den niedrigen Energieverbrauch aus. Der Rauschabstand ist laut Datenblatt 94 dB bei 1 kHz, die Total Harmonic Distortion (THD) bei 1%. Das Mikrofon erscheint damit als sehr geeignet. Das digitale Format, ausgegeben über I2S, hat 24 bit, wobei nur 18 verwendet werden. Die Software über die der Datenstrom dann schließlich verarbeitet wird nimmt diesen allerdings mit 32 Bit auf. Es erscheint daher sinnvoll, in Software die Bittiefe gleich im Anschluss an jede Aufnahme auf 24 oder 16 Bit (16 Bit entspricht der Auflösung bei einer Audio-CD, 18 Bit ist keiner der gängigen unterstützten Werte der verwendeten Audioformate) herabzusetzen, um Speicherplatz und Rechenkapazität zu sparen. Die Mikrofone sind am Foto in Abbildung 4.5 gezeigt.

4.1.7. Speicherkarte

Der RPI unterstützt als Hauptspeicher SD-Karten im micro Format. Eine günstige Karte genügt, da die laufenden Operationen ohnehin im Arbeitsspeicher durchgeführt werden. Es fallen täglich etwa 100 MB an Daten an, eine 16 GB Speicherkarte kann also Aufnahmen für etwa ein halbes Jahr speichern.

4. Wahl der Hard- und Software

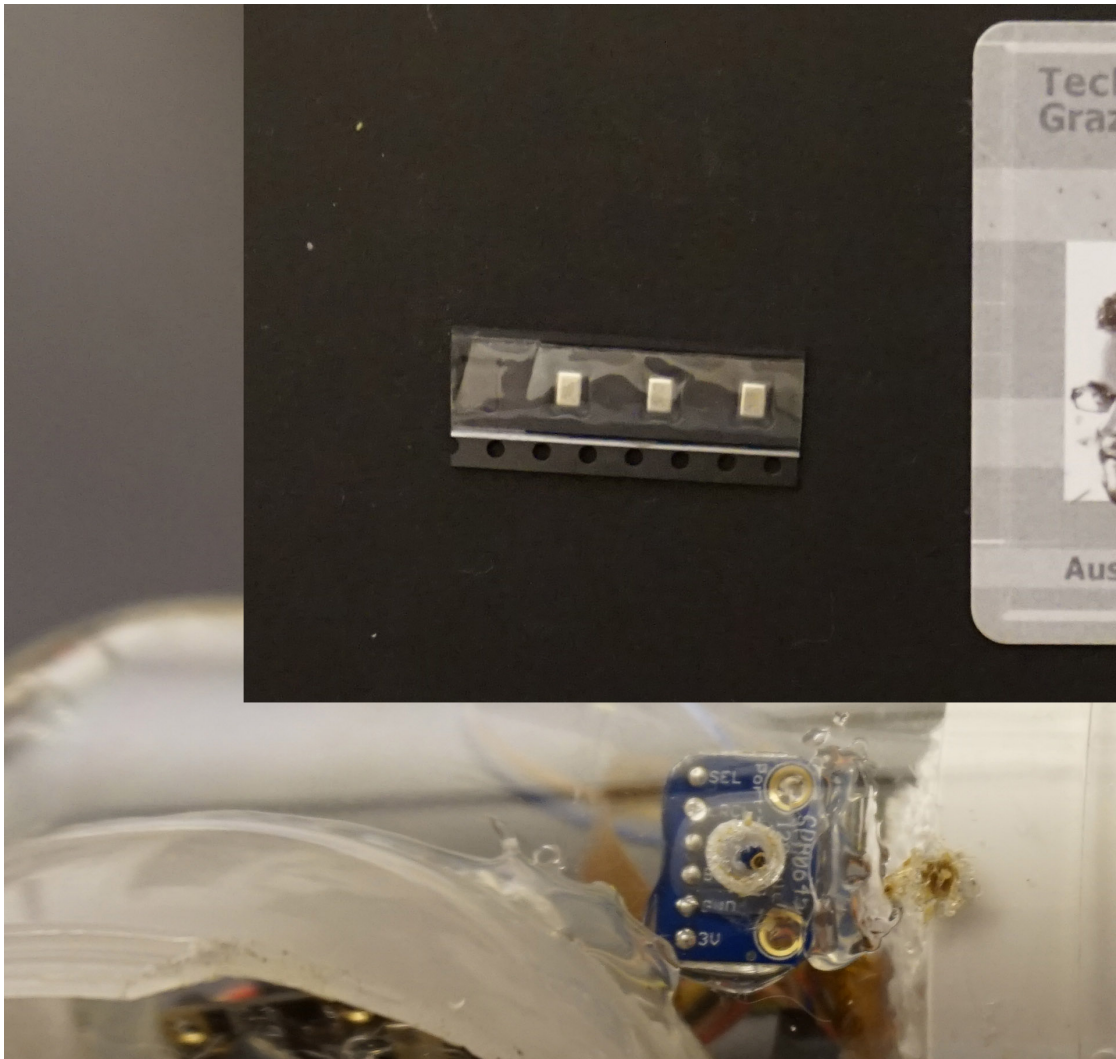


Abbildung 4.5.: RPI Mikrofon: Oben 3 Mikrofone, verpackt, unten als Modul und eingebaut

4.2. Software

In diesem Kapitel werden die wichtigsten verwendeten Softwarepakete beschrieben. Dazu werden jeweils die Anforderungen und mögliche Alternativen genannt, sowie abschließend die erfolgte Wahl begründet. Über alle Komponenten hinweg sind Lizenz (Open Source) und Kompatibilität zum RPI relevante Aspekte.

4.2.1. Betriebssystem

Es existieren zahlreiche linuxbasierte Betriebssysteme die am RPI lauffähig sind oder speziell für die Plattform entwickelt oder angepasst wurden. Für das Projekt ist eine Variante ohne graphische Oberfläche (headless) vorzuziehen: Für die Funktion ist keine Oberfläche notwendig, die knappen Ressourcen werden davon nicht belastet. Prinzipiell kommen als Alternative zu Linux auch FreeBSD/ARM¹³, Android¹⁴ oder Windows 10 IoT Core¹⁵ infrage. Mit der Minimalversion von Raspbian existiert aber ein Betriebssystem, das direkt vom Hersteller des RPI zur Verfügung gestellt wird und auf Debian, einer der ältesten und am weitesten verbreiteten Linuxdistributionen basiert. Die Wahl fällt daher auf Raspbian Stretch Lite, für das auch die allermeisten Softwarelösungen, etwa zum Ansteuern weiterer Hardware wie der Kamera und den Mikrofonen, vorliegt.

Zwei für das Projekt wichtige Möglichkeiten, die das Linux Betriebssystem anbietet sind die Möglichkeit Dateien (temporär) im Arbeitsspeicher, in einer Ramdisk, abzulegen und mit sogenannten named pipes arbeiten zu können:

Eine **Ramdisk** wird unter Linux mit einem einfachen Eintrag in den file systems table (fstab) angelegt. Dazu wird eine maximale Größe angegeben. Bis zu dieser wird vom System dann Arbeitsspeicher bereit gestellt, in den Dateien ganz normal abgelegt werden können. Das bringt zwei Vorteile, die für das Projekt Voraussetzung sind: Einerseits wird so eine Abnutzung der SD-Karte, die nicht auf viele Schreibzyklen ausgelegt ist, vermieden. Andererseits sind Operationen auf Dateien die im Arbeitsspeicher liegen ungleich schneller, wie ein einfacher Test am RPI eindrucksvoll zeigt: Der Befehl `dd if=/dev/zero of=output bs=8k count=50k` schreibt in 50k Blöcken zu je 8 Kb die von /dev/zero gelesenen Daten in eine Datei 'output' und erreicht in der Ramdisk knapp 300 MB/s, auf der SD-Karte hingegen nur gut 10 MB/s. Auf die Ramdisk kann

¹³siehe <https://wiki.freebsd.org/FreeBSD/arm/Raspberry%20Pi>, abgerufen 18-04-24

¹⁴siehe http://androidpi.wikia.com/wiki/Android_Pi_Wiki, abgerufen 18-04-24

¹⁵siehe <https://developer.microsoft.com/en-us/windows/iot/downloads>, abgerufen 18-04-24

4. Wahl der Hard- und Software

also gut 25 mal so schnell geschrieben werden wie auf die SD-Karte. Die Geschwindigkeit beim Lesen ist weniger relevant, dabei ist die SD-Karte mit 23 MB/s etwas schneller.¹⁶

```
1 root@pi ~ # dd if=/dev/zero of=output bs=8k count=50k
3 51200+0 records in
5 51200+0 records out
7 419430400 bytes (419 MB, 400 MiB) copied, 38.7671 s, 10.8 MB/s
9
11 root@pi ~ # dd if=/dev/zero of=/tmp2/output bs=8k count=10k
13 10240+0 records in
15 10240+0 records out
17 83886080 bytes (84 MB, 80 MiB) copied, 0.320998 s, 261 MB/s
19
21 root@pi ~ # dd if=/dev/zero of=/tmp2/output bs=8k count=10k
23 10240+0 records in
25 10240+0 records out
27 83886080 bytes (84 MB, 80 MiB) copied, 0.323026 s, 260 MB/s
29
31 root@pi ~ # dd if=/dev/zero of=/tmp2/output bs=8k count=10k
33 10240+0 records in
35 10240+0 records out
37 83886080 bytes (84 MB, 80 MiB) copied, 0.285661 s, 294 MB/s
39
41 root@pi ~ # nice -n -20 dd if=/dev/zero of=/tmp2/output bs=8k count=10k
43 10240+0 records in
45 10240+0 records out
47 83886080 bytes (84 MB, 80 MiB) copied, 0.28862 s, 291 MB/s
49
51 root@pi ~ # nice -n -20 dd if=/dev/zero of=/tmp2/output bs=8k count=10k
53 10240+0 records in
55 10240+0 records out
57 83886080 bytes (84 MB, 80 MiB) copied, 0.291103 s, 288 MB/s
```

Listing 4.1: Benchmark SD-Karte vs Ramdisk

Mittels **Named Pipes** können Schreibvorgänge in (reguläre) Dateien ganz vermieden werden. Dazu wird eine spezielle Datei am System angelegt die sich als FIFO Speicher verhält. Ein Prozess schreibt in diese Datei, danach kann ein anderer Prozess die geschriebenen Daten auslesen. Für das Projekt wartet ffmpeg als lesender Prozess auf neue Bilder, die nach und nach von der Kamera aufgenommen, in der Ramdisk abgelegt und in die Named Pipe geschrieben werden. Bei diesem Vorgang verlassen die Daten also nie den Arbeitsspeicher und schonen damit die SD-Karte.

¹⁶In der unter /tmp2/ eingehängten Ramdisk können, da der RPI Zero insgesamt nur über 512 MB Ram verfügt, nicht 400 MB auf einmal zum Testen geschrieben werden - daher wird der Test hier fünf mal mit 80 MB wiederholt.

4. Wahl der Hard- und Software

Raspbian kann von der Website des Herstellers bezogen werden¹⁷ und besteht aus Komponenten die allesamt FOSS entsprechen, hauptsächlich GPL. Programme, die unter nicht-freien Lizenzen stehen sind in der verwendeten Minimalversion nicht vorhanden.

4.2.2. Skriptsprache

Im Vergleich zu beispielsweise C sind Skriptsprachen wie Python wesentlich einfacher zugänglich. Es fallen viele Aspekte wie Speicherverwaltung weg und damit lassen sich schneller Ergebnisse erarbeiten. Daher soll das Projekt mit dem Einsatz von Skripten auskommen, die einfach - also insbesondere ohne Verständnis von Pointern, Speicherverwaltung oder Objektorientierung - nachvollzogen und adaptiert werden können. Da die rechenintensiven Operationen ohnehin von hoch optimierten Programmen, die selbst wie ffmpeg etwa in C und Assembler¹⁸ geschrieben sind, vorgenommen werden, ergibt sich auch kein relevanter zusätzlicher Ressourcenverbrauch. Unter Raspbian unterstützte und gängige Sprachen sind beispielsweise Scratch, Bash, Python und Perl. Die Wahl fällt auf Bash, das bereits seit Jahrzehnten auf Linux Anwendung findet und entsprechend gut dokumentiert und integriert ist. Auch bei Bash handelt es sich um FOSS lizenzierte Software, die bereits mit Raspbian vorinstalliert ausgeliefert wird.

4.2.3. Kamera

Zur Verwendung der Kamera ist unter Raspbian raspistill¹⁹ installiert. Auch von dieser Software ist der Quellcode verfügbar und darf verändert werden.

4.2.4. Video

Die größten und aktiv weiterentwickelten Linux Softwarepakete zur Transkodierung von Videodaten sind mencoder, handbrake, ffmpeg beziehungsweise ein Fork davon namens libav. Diverse weitere Programme, beispielsweise avidemux, stellen lediglich eine grafische Oberfläche für eines dieser Softwarepakete bereit. Davon unterstützt lediglich ffmpeg das Lesen von einer Folge von Bildern via Pipe. Daher fällt die Wahl auf ffmpeg.

¹⁷unter <https://www.raspberrypi.org/downloads/raspbian/>, abgerufen 18-04-24

¹⁸vgl. <http://ffmpeg.org/developer.html>, abgerufen 18-04-24

¹⁹Dokumentation unter <https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md>, abgerufen 18-04-24

4.2.5. Audio

Es existieren diverse Programme, die Audiodaten von einer Quelle, etwa einem Mikrofon, aufnehmen können. Andere, auch hier ist ffmpeg wieder ein Beispiel, können diverse Transkodierungen in verschiedene Formate durchführen. Insgesamt verfügt aber SoX über den größten Funktionsumfang, insbesondere die splice-Funktion wird im Projekt verwendet.

4.3. Bild-, Audio- und Videoformate

In diesem Kapitel werden grundlegende Eigenschaften von Audio- und Videoformaten und Containern beschrieben und einige in Frage kommende Formate verglichen. Anschließend wird die Entscheidung für jedes schlussendlich verwendete Format begründet.

4.3.1. Bild

Für digitale Bilddateien sind Auflösung, Farbtiefe, Kompression und Kompressionsfaktor die wichtigsten Eigenschaften.

- Die **Auflösung** bestimmt das Seitenverhältnis des Bildes (nicht-rechtwinklige Formate sind unüblich) und die Anzahl der Bildpunkte.
- Die **Farbtiefe** sagt aus, mit welcher Genauigkeit die Farbe eines einzelnen Pixels gespeichert wird. Bei 8 Bit RGB wird die Farbe aus je einem Anteil an Rot, Grün und Blau zusammengesetzt und jeder dieser Farbkanäle mit 8 Bit Genauigkeit angegeben. Damit können jeweils 256 unterschiedliche Intensitäten ausgedrückt und somit gut 16 Millionen unterschiedliche Farben dargestellt werden. Ohne weitere Komprimierung hat damit ein Bild mit einer Auflösung von 1920 mal 1080 Bildpunkten knapp 6 MB.
- Eine **Komprimierung** kann entweder verlustbehaftet oder verlustfrei erfolgen. Im Fall von verlustbehafteten Komprimierungsverfahren wie JPEG kann das Originalbild nicht mehr perfekt rekonstruiert werden. Je nach Kompressionsfaktor sind die Unterschiede mehr oder weniger stark sichtbar. Es existieren auch verlustfreie Verfahren, beispielsweise PNG, die Redundanzen im Bildmaterial ausnutzen, um einen sparsamen Weg der Speicherung zu finden.

Um ohne unnötige Konvertierungen auszukommen sollte ein Bildformat verwendet werden, dass von raspistill und ffmpeg gleichermaßen unterstützt wird. Außerdem sind Performanz und Qualität die wichtigen Kriterien.

4. Wahl der Hard- und Software

Von `ffmpeg` werden sowohl `jpeg_pipe` (piped jpeg sequence) als auch `png_pipe` (piped png sequence) als Eingangsformate unterstützt. Das Bild selbst kann von `raspistill` ebenso unter anderem als JPEG und PNG aufgenommen werden. Ein einfacher Test zwischen verlustfreiem PNG und beinahe verlustfreiem JPEG mit einem Qualitätsfaktor von 100% zeigt warum die Wahl auf JPEG fällt:

```
1 root@pi ~ # time nice -n -19 /usr/bin/raspistill --thumb none -ex night -a
   1028 -ae 15,0x00,0x8080FF -a "%Y-%m-%d %X" -w 1920 -h 1080 -n -o /tmp2/
   test.jpg -e jpg -q 100
3 real    0m5.685s
   user    0m0.012s
5 sys     0m0.061s
7 root@pi ~ # time nice -n -19 /usr/bin/raspistill --thumb none -ex night -a
   1028 -ae 15,0x00,0x8080FF -a "%Y-%m-%d %X" -w 1920 -h 1080 -n -e png -o /
   tmp2/test.png
9 real    0m8.840s
   user    0m0.017s
   sys     0m0.102s
11 root@pi ~ # ls -lh test*
13 -rw-r--r-- 1 root root 1.5M Mar 31 18:05 test.jpg
   -rw-r--r-- 1 root root 3.3M Mar 31 18:08 test.png
```

Listing 4.2: Benchmark jpg versus png

Dabei ist das Problem nicht so sehr, dass die als PNG gespeicherte Datei mehr als doppelt so groß ist, sondern vor allem, dass `raspistill` beim Speichern als PNG länger braucht. Bei langen Belichtungszeiten könnten damit die gewünschten 15 Sekunden Intervalle vielleicht nicht immer eingehalten werden. Auch bei PNG gibt es verschiedene Kompressionsstrategien, aus denen gewählt werden kann - mit der Geschwindigkeit von JPEG kann aber keine der Varianten mithalten.

Daher fällt die Wahl auf JPEG als Dateiformat für die Bilder.

4.3.2. Audio

Für digitale Audiodateien sind Abtastrate, Bittiefe, Anzahl der Kanäle, Bitrate und Komprimierung die wichtigsten Eigenschaften.

Die **Abtastrate** sagt aus, wie häufig ein Signal abgetastet, also gemessen wird. Im Fall eines Audiosignals und der Aufnahme durch ein Mikrofon bedeutet das, dass mit dieser Häufigkeit pro Sekunde die durch die Bewegung der Membran in der Spule des Mikrofons induzierte Spannung gemessen wird. Gemäß Abtasttheorem muss dies mit mehr als der doppelten Frequenz der höchsten im Signal vorkommenden Frequenz

4. Wahl der Hard- und Software

geschehen, um das Signal eindeutig rekonstruieren zu können. Für Audiosignale, die für das menschliche Gehör bestimmt sind, genügen daher Abtastraten im Bereich von 40 kHz. Audio-CDs verwenden 44.1 kHz. Die für das Projekt verwendeten Mikrofone unterstützen laut Datenblatt²⁰ Abtastraten von bis zu 64 kHz. Da höhere Abtastraten zu entsprechend höherem Ressourcenbedarf in der weiteren Verarbeitung und Speicherung führen, wird 48 kHz gewählt, wie es dem aktuellen Industriestandard²¹ im Video- und zunehmend auch Audiobereich entspricht.

Die **Bittiefe** bestimmt, wie fein aufgelöst ein einzelner Messwert ist. 16 Bit, wie bei einer Audio-CD üblich, entsprechen 65536 verschiedenen möglichen Werten. Von den Mikrofonen werden Signale mit einer Auflösung von 18 Bit aufgenommen, als 24 Bit Datenstrom ausgegeben und am RPI als 32 Bit Strom eingelesen. Da höhere Auflösungen zu entsprechend höherem Ressourcenbedarf in der weiteren Verarbeitung und Speicherung führen, wird gleich im Anschluss an die Aufnahme auf 16 Bit umgewandelt.

Prinzipiell ist eine beliebige **Anzahl der Kanäle** denkbar. Die verwendeten Mikrofone unterstützen Mono- und Stereoaufnahmen. Für letzteres werden die Mikrofone dem linken beziehungsweise rechten Kanal zugewiesen, indem ein Pin an die Versorgungsspannung angeschlossen wird. Für das Projekt werden 2 Mikrofone und entsprechend Stereo, also 2 Kanäle verwendet.

Die **Bitrate**, also wie viel Speicherbedarf pro Zeit anfällt, ergibt sich im Fall von nicht-komprimierten Formaten aus den genannten Eigenschaften und zusätzlich etwaigem Overhead durch das jeweilige Dateiformat und Metadaten. Bei den gewählten 48 kHz und 16 Bit auf 2 Kanälen sind es knapp 200 KB pro Sekunde oder gut 10 MB je Minute.

Wie bei Bildern kann eine **Komprimierung** entweder verlustbehaftet oder verlustfrei erfolgen. Im Fall von verlustbehafteten Komprimierungsverfahren (am bekanntesten ist sicher nach wie vor MP3, andere bekannte und aktuelle Formate sind Ogg Vorbis, Opus und AAC) findet eine Datenreduktion statt und es kann üblicherweise die Bitrate vorgegeben werden, an der sich dann die Qualität des Ergebnisses orientiert. Alternativ erlauben manche Formate eine gewünschte Qualität vorzugeben. Je nachdem, wie das zur Komprimierung vorgesehene Audiosignal aufgebaut ist ergibt sich daraus dann der Speicherbedarf. Im Fall von verlustfreier Komprimierung (etwa mit FLAC oder WavPack) findet eine Datenkompression statt, die darauf basiert eine sparsamere Repräsentation der Daten zu finden - beispielsweise indem Ähnlichkeiten zwischen den Kanälen oder den aufeinanderfolgenden Abtastwerten gefunden werden.

²⁰verfügbar unter <http://www.knowles.com/eng/content/download/5988/105771/version/4/file/SPH0645LM4H-BRevB.PDF>, abgerufen 18-04-24

²¹Siehe z. B. die Argumente unter www.provideocoalition.com/all-audio-production-distribution-should-go-48-khz-learn-why/, abgerufen 18-04-24

4. Wahl der Hard- und Software

Für das Projekt wird mit unkomprimierten Audiodateien, verlustfreier und verlustbehafteter Kompression gearbeitet. Dabei werden die Formate so eingesetzt, dass keine unnötigen Qualitätsverluste während den Verarbeitungsschritten entstehen. Erst ganz am Ende wird das verlustfreie Format in ein verlustbehaftetes umgewandelt. Verwendung finden die Formate PCM, gespeichert im Waveform Audio Dateiformat, FLAC für verlustfreie Kompression und OGG Vorbis für verlustbehaftete Kompression.

4.3.3. Video

Für digitale Videodaten sind Bildrate, Auflösung, Farbtiefe, Komprimierung und Bitrate die wichtigsten Eigenschaften.

- Die **Bildrate** gibt an, aus wie vielen Einzelbildern pro Zeiteinheit das Video zusammengesetzt ist. Im Projekt wird alle 15 Sekunden ein Bild aufgenommen, also 5760 Bilder am Tag. Bei einer Bildrate von 60 Bildern pro Sekunde ergibt sich daraus eine Länge des Videos von 96 Sekunden.
- Die **Auflösung** bestimmt wie beim Bild das Seitenverhältnis des Videos (nicht-rechtwinklige Formate sind unüblich) und die Anzahl der Bildpunkte. Häufig verwendete Auflösungen tragen eigene Namen, die ausgewählte Auflösung von 1920 mal 1080 Bildpunkten ist beispielsweise als FullHD, oder im Falle der Verwendung von Vollbildern (der Einsatz von Halbbildern bei doppelter Frequenz ist ein Überbleibsel aus analogen Formaten für das Fernsehen wie PAL und NTSC) als 1080p bekannt. Häufig wird außerdem die Bildrate angegeben, damit entspricht das Format dem Kürzel 1080p60.
- Die **Farbtiefe** sagt aus, mit welcher Genauigkeit die Farbe eines einzelnen Pixels gespeichert wird. Sehr verbreitet sind Farbtiefen von 8 bit, wobei aktuell auch zunehmend Bildschirme Verbreitung finden, die mit dem Kürzel HDR werben, was für eine höhere Farbtiefe von 10 Bit steht. Im Projekt wird, wenn nicht anders angegeben, eine Farbtiefe von 8 Bit verwendet.

Ein Speichern ohne Komprimierung von Videomaterial ist auf einem RPI aufgrund der extrem hohen Datenraten praktisch ausgeschlossen. Für die geplanten 96 Sekunden Video wären etwa 35 GB notwendig, bei einer Bitrate von 373 MB/s. Auch verlustfreie Kompressionsverfahren erreichen nicht die notwendige Reduktion, daher kommt nur eine verlustbehaftete Variante in Frage. Gängig sind dazu aktuell beispielsweise H.264/MPEG-4 AVC, H.265 / HEVC, Theora oder VP9. Zwar liefert H.265 bessere Ergebnisse (höhere Qualität bei weniger Speicherbedarf), hat aber höhere Anforderungen an die Hardware und wird noch von weniger Hard- und Software unterstützt. Theora und VP9 haben den Vorteil, jeweils ein lizenzgebührenfreies, offenes Format beziehungsweise patentfreies Verfahren zu sein. Seit 2010 handelt es sich auch bei H.264

4. Wahl der Hard- und Software

um ein für private Nutzung lizenzgebührenfreies Format²². Aufgrund der großen Verbreitung und Unterstützung insbesondere auch in Hardware wird H.264 für dieses Projekt genutzt, in Zukunft könnte der aktuell von der Alliance for Open Media²³ entwickelte AV1 Standard eine bessere, offene Alternative darstellen.

4.3.4. Container

Ein Containerformat fungiert als 'Behälter' für (verschiedene) Datenströme, die in der Containerdatei zusammengeführt werden - das sind beispielsweise (mehrere) Video- und Tonspuren. Eine wichtige Aufgabe dabei ist es, sicherzustellen, dass die Datenströme synchron abgespielt werden können. Als Containerformat kommen jene Varianten infrage, die die gewünschten Audio- und Videoformate unterstützen und mit ffmpeg verwendet werden können. Das schränkt die Wahl auf folgende Formate ein: Matroska, MXF (Material eXchange Format), Ogg und QuickTime / MOV

An QuickTime hält Apple die Rechte, bei den anderen Formaten handelt es sich um offene Standards. MXF wurde für professionelle Anwendungen entwickelt. Ogg ist lediglich als Audiocontainer weiter verbreitet, als Containerformat wird daher Matroska (.mkv) gewählt.

²²siehe <http://www.mpegla.com/Lists/MPEG%20LA%20News%20List/Attachments/74/n-10-08-26.pdf>, abgerufen 18-04-24

²³<https://aomedia.org>

5. Zusammenfassung

Im ersten Teil dieser Arbeit wurde in wichtige Grundlagen für das Projekt eingeführt und die Entscheidung für Anforderungen, Hard- und Software, Bild-, Audio- und Videoformate motiviert:

- Anforderungen:
 - Robustheit - Neustart im Fall von Fehlern, Stromausfällen etc.
 - Verwendung von freier, quelloffener Software (FOSS)
 - FullHD Auflösung mit 60 Bildern pro Sekunde
 - mehrere Aufnahmen pro Minute
- Hardware:
 - RPI Zero W
 - Weitwinkel RPI Camera
 - Geekworm USV
 - 18650 Panasonic LiPo Akku
 - 2 I2S SPH0645LM4H Mikrofone
- Software:
 - Raspbian
 - Bash
 - raspistill
 - ffmpeg
 - SoX
- Formate:
 - Bild: JPEG
 - Audio: 16 Bit PCM WAV, FLAC, OGG Vorbis
 - Video: H.264
 - Container: Matroska

Teil II.

Umsetzung

Einleitung

Im zweiten Teil dieser Arbeit wird die Umsetzung des Projektes dokumentiert. Das Ziel ist dabei zwar keine grundlegende Einführung in die Arbeit mit Linux oder dem RPI generell zu geben, aber doch ausreichend detailliert alle notwendigen Schritte zu beschreiben, um das Projekt mit angemessenem Aufwand nachbauen und die Ergebnisse reproduzieren zu können. Dazu wird zuerst die grundlegende Konfiguration des RPI beschreiben, dann die Hardwareseite betrachtet und schließlich die Software für die eigentliche Zeitrafferfunktion erklärt. Außerdem werden verschiedene Varianten, die erprobt und verworfen wurden, beschrieben und Erweiterungen vorgestellt, die über die Zeitrafferaufnahme (aber auch über die Möglichkeiten des RPI Zero) hinausgehen. Die Ergebnisse werden danach im dritten Teil der Arbeit präsentiert und analysiert.

6. Konfiguration

Ausgegangen wird von einer frisch installierten Version Raspbian Stretch (Kernel Version 4.9). Anleitungen dazu finden sich auf der RPI Website¹.

6.1. Einrichtung, Aktualisierungen, Installation von Softwarepaketen

Diverse Einstellungen können einfach mit dem tool `raspi-config` vorgenommen werden, etwa das Anpassen der Partitionstabellen, so dass die volle Größe der SD-Karte genutzt wird. Zu Beginn sollte das System aktualisiert werden, dazu gehört beim RPI auch ein Update der Firmware. Dazu eignen sich die folgenden Befehle, jeweils als root oder mit `sudo` ausgeführt:

¹unter <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>, abgerufen 18-04-24

6. Konfiguration

```
aptitude update && aptitude upgrade && rpi-update
```

Folgende Programme sind unter Raspbian nicht standardmäßig installiert und sollten jeweils als root oder mittels sudo via "aptitude install Programmname" installiert werden:

- **bc** erlaubt Berechnungen auch mit Fließkomma, wird in mehreren bash Skripten eingesetzt.
- **bmon** zeigt die Auslastung des Netzwerks an, wichtig zum Debugging von Netzwerkproblemen.
- **ffmpeg** übernimmt das Zusammensetzen der einzelnen Bilder und komprimiert das so entstehende Video. Wird auch genutzt um das entstandene Video am Ende mit der Tonspur zusammenzufügen und diese neu zu komprimieren.
- **izc-tools** ist notwendig um den Ladezustand des Akkus an der USV, die mittels I2C kommuniziert, auszulesen.
- **nginx** ist ein im Vergleich zu apache sehr ressourcenschonender Webserver. Wird verwendet um Dateien einfach via Webbrowser vom RPI herunterzuladen.
- **SoX** übernimmt die Aufnahme, Verarbeitung und verlustfreie Kompression der Zeitraffertonaufnahme.
- **stress-ng** lastet die CPU aus, notwendig für einen ersten burn-in.
- **tmux** erlaubt es, ähnlich wie das bekanntere aber weniger umfangreiche Programm screen Programme weiterlaufen zu lassen, ohne dass eine SSH-Verbindung aufrechterhalten wird.

6.2. BIOS-Konfiguration

Die RPI-Modelle nutzen allesamt keine übliche BIOS-Oberfläche sondern eine einfache Textdatei, die beim Start eingelesen wird. Folgende Parameter werden vom Standard abweichend gesetzt:

- **hdmi_force_hotplug=1** - Erlaubt im Fehlerfall auch ohne Neustart einen Monitor via HDMI anzuschließen. Ohne diese Einstellung wird ein Display erst bei einem Neustart erkannt.
- **dtparam=i2c_arm=on** - Auf diese Art wird das i2s Hardwareinterface aktiviert, über das die USV mit dem RPI kommuniziert.
- **dtparam=i2s=on** - Über den i2s Standard können digital Audiodaten übertragen werden. Er wird von den beiden verwendeten Mikrofonen verwendet.
- **dtparam=audio=on** - Über diesen Parameter wird eingestellt, dass das Audio-Modul `snd_bcm2835` geladen wird, das ist Voraussetzung für die Tonaufnahmen

6. Konfiguration

- `start_x=1` - Hiermit wird die über das DSI Interface angeschlossene Kamera aktiviert.
- `gpu_mem=64` - Dieser Parameter setzt fest, wie viel von den 512MB Arbeitsspeicher des RPI Zero W für die Grafik zur Verfügung steht. Zum Betrieb der Kamera sind eigentlich 128MB Voraussetzung, im Fehlerfall weist darauf auch die Fehlermeldung hin: Wenn auf Videoaufzeichnungen verzichtet wird und, wie geplant nur Fotoaufnahmen in einer Auflösung von 1920 mal 1080 gemacht werden, reichen 64MB allerdings aus.
- `disable_camera_led=1` - Schaltet das - für nächtliche Aufnahmen potentiell wegen Spiegelungen störende - rote Led aus.

6.3. Auslagerungsdatei

Innerhalb eines kurzen Zeitraumes am Übergang von einem Tag zum nächsten wird eine Aufnahme beendet und eine zweite parallel gestartet. Damit ist sichergestellt, dass zu jedem vorgesehenen Moment auch eine Aufnahme erstellt wird. Dazu ist es notwendig, dass zwei Instanzen von ffmpeg gleichzeitig laufen während auch noch die gesamten Aufnahmen des Tages in der Ramdisk, also auch im Arbeitsspeicher, liegen. An diesem Punkt entsteht die höchste Auslastung des Arbeitsspeichers. Um zu vermeiden, dass der Kernel Prozesse beendet um Speicher freizugeben wird Swapping aktiviert und eine entsprechend große Auslagerungsdatei von 1GB vorgesehen.

Um zu vermeiden dass der Kernel zu häufig beginnt, Daten aus dem Arbeitsspeicher in die Auslagerungsdatei zu schieben, wird der Parameter `swappiness` von 60 auf 20 herabgesetzt. Insgesamt ist besonders auf einem RPI die Verwendung der Auslagerungsdatei zwar in einem Fall wie diesem notwendig, aber möglichst einzuschränken: Einerseits sinkt die Leistung sehr stark, da die Schreib- und Leserraten auf der als Speicher genutzten SD-Karte im Vergleich zu Festplatten oder SSDs sehr niedrig sind, andererseits ist auch mit einer massiven Lebensdauerverkürzung zu rechnen, wenn regelmäßig große Datenmengen in die Auslagerungsdatei und damit auf die Karte geschrieben werden. Die Einstellungen werden auf folgende Weise vorgenommen:

- Konfigurationsdatei: `/etc/dphys-swapfile`
- Einstellung: `CONF_SWAPFILE=1024`
- Befehl: `sysctl vm.swappiness=20`
- Konfigurationsdatei: `/etc/sysctl.conf`
- Einstellung: `vm.swappiness = 20`

6.4. Wireless Lan

Der RPI Zero W verfügt zusätzlich zum RPI Zero über WLAN und Bluetooth. Um die Verbindung mit einem Netzwerk herzustellen sind folgende Einstellungen vorzunehmen:

Konfigurationsdatei: `/etc/wpa_supplicant/wpa_supplicant.conf`

```
1 country=AT
network={
3     ssid="SSID"
     psk="Password"
5 }
```

6.5. Mikrofone einrichten

Zum Einrichten der Mikrofone muss unter anderem ein Kernelmodul compiliert werden. Eine entsprechende Anleitung würde den Rahmen dieser Arbeit sprengen, es sei daher auf die Hinweise auf der Website der Herstellerin verwiesen².

²unter <https://learn.adafruit.com/adafruit-i2s-mems-microphone-breakout/>, abgerufen 18-04-24

7. Zeitrafferaufnahmen am RPI

In diesem Kapitel wird ein trivialer Ansatz zur Erstellung des Zeitraffervideos, eine Verbesserung und schließlich die optimale Variante, die auch weiter verfolgt wurde, beschrieben. Es wird gezeigt wie Zeitraffertonaufnahmen umgesetzt werden, wie der RPI überwacht werden kann, und es werden einige Möglichkeiten für Erweiterungen präsentiert, die mitunter über die Möglichkeiten des RPI Zero hinausgehen.

7.1. Verschiedene Lösungsansätze

Der triviale Ansatz ist zugleich jener, der in den meisten Anleitungen zur Erstellung von Zeitrafferaufnahmen erklärt wird (siehe erster Teil der Arbeit). Dazu werden zuerst, während der gesamten Aufnahmedauer, alle Einzelbilder gespeichert und erst im Anschluss zu einem Video weiterverarbeitet. Diese Variante bietet Vorteile, wenn ohnehin eine manuelle Nachbearbeitung der Bilder und des entstehenden Videos vorgesehen ist. In diesem Fall liegen auch am Ende der Aufnahme noch alle Fotos in voller Qualität zur Bearbeitung vor. Für eine automatisierte Verarbeitung auf einem RPI ist die Variante aber ungeeignet, da während der langen Phase der Aufnahme die verfügbare Rechenleistung kaum ausgenutzt wird. Stattdessen entstehen Spitzenlasten im Anschluss an die Aufnahme. Außerdem wären die Speicheranforderungen sehr hoch: 5760 Bilder, jeweils etwa 1-2 MB würde mit bis zu 10GB den verfügbaren Arbeitsspeicher deutlich übersteigen. Eine Zwischenspeicherung auf der SD-Karte wiederum würde die im einführenden Kapitel beschriebenen Nachteile mit sich bringen. Eine bessere Lösung nutzt mehr der verfügbaren Rechenleistung, vermindert Spitzenauslastungen und hat geringere Anforderungen an Speicherplatz.

Verbesserter Ansatz

Das Intervall in dem gemäß dem trivialen ersten Ansatz ein Video aus den einzelnen Bildern erstellt wird kann beliebig kurz gewählt werden. Im Extremfall von nur wenigen oder überhaupt nur einem Bild ist das Ziel eines Zeitraffervideos natürlich nicht mehr erfüllt. Kürzere Videosegmente können allerdings ohne Probleme und ohne, dass das Videomaterial erneut rechenintensiv transkodiert werden muss, aneinandergesetzt

7. Zeitrafferaufnahmen am RPI

werden. Wird allerdings ein Video aus nur einem Bild erstellt¹ und werden diese Videos dann aneinander gereiht, so geht jede Möglichkeit verloren, eine Datenreduktion durch Ausnutzung der Ähnlichkeiten von aufeinander folgenden Bildern zu bewirken. Es ist also entsprechend dieser entgegengesetzten Anforderungen eine Abwägung zu treffen. Es liegt nahe, zumindest etwa so viele Bilder zu verwenden wie es der Anzahl an Bildern entspricht, die der verwendete Codec in die Zukunft beziehungsweise in die Vergangenheit schaut, um eine Komprimierung zu erreichen.

Der Ablauf erfolgt dann zyklisch wie folgt:

1. Laufende Aufnahme von Bildern unter laufender Nummerierung, beispielsweise nach Uhrzeit.
2. Sobald n Bilder aufgenommen sind, werden sie in einen Unterordner verschoben.
3. Dann startet ffmpeg und setzt diese n Bilder zu einem Video zusammen, das ebenfalls mit der Uhrzeit als Dateiname gespeichert wird.
4. Der Ordner mit den n Bildern wird im Anschluss gelöscht. In der Zwischenzeit dürfen maximal wieder n neue Bilder entstanden sein - ansonsten müssen Schritte gesetzt werden, um die Anforderungen an die Rechenleistung zu senken.

Der Speicherplatzbedarf entspricht also $2n$ -mal der durchschnittlichen Größe eines Bildes, dazu kommen die Teil-Videos. Bei $n = 60$ (also Teilvideos zu je einer Sekunde) ergibt das insgesamt etwa (2 MB pro Bild, 40 MB für das Video) 140 MB. Da am Ende auch noch das Gesamtvideo aus den Einzelteilen zusammengesetzt werden muss steigt der Spitzenbedarf auf etwa 180MB. Am RPI Zero sind insgesamt nur 512 MB Ram vorhanden, 448 bleiben nach Abzug des Speichers, den der Videochip benötigt und ffmpeg selbst braucht zwischen 200 und 300 MB. Das Betriebssystem selbst sowie weitere Programme brauchen natürlich auch entsprechend viel Arbeitsspeicher, was nahe legt, eine weitere Verbesserung zu finden.

Ein weiterer Nachteil zeigt sich, wenn durch Unterschiede im Bildmaterial die Rechenzeit für ffmpeg steigt: Dann kann es vorkommen, dass ein eigentlich ausreichend langes Intervall doch nicht zur Bearbeitung ausreicht. Für diesen Fall ist natürlich eine entsprechende Fehlerbehandlung vorzusehen.

Der dritte Teil der Arbeit zeigt wie diese Variante auf einem RPI 2 mit 1 GB Arbeitsspeicher sehr gut funktioniert - für den RPI Zero ist sie eher ungeeignet.

Optimale Lösung

Bei der optimalen Lösung müssen keine Bilder zwischengespeichert werden. Das funktioniert, indem die Aufnahmen nach und nach ffmpeg zugeführt werden. Das

¹Auch das kann sinnvoll sein, wenn der Videocodec auch Einzelbilder schon sehr gut komprimiert - vgl. <http://nokiatech.github.io/heif>, abgerufen 18-04-24

7. Zeitrafferaufnahmen am RPI

gelingt, da `ffmpeg` als Eingangsformat `jpeg_pipe` unterstützt, also JPEG-Bilder aus einer pipe lesen kann. Ein Befehl dazu könnte im einfachsten Fall also etwa so aufgebaut sein, mit `programm_a` als Quelle, also etwa `raspistill`, und `programm_b` als Senke, also etwa `ffmpeg`:

```
programm_a | programm_b
```

Dazu müsste natürlich noch die Ausgabe der Bilddaten auf `stdout` umgeleitet werden. Bei dieser Variante würde `ffmpeg` aber nach jedem Bild ein EOF empfangen, das Video schließen und sich beenden. Es ist aber schon deshalb keine Lösung, weil so ja beide Programme mit jedem Bild neu gestartet und beendet werden, womit kein durchgängiges Video entstehen würde.

Die Lösung besteht darin, statt `|`, also einer Unnamed Pipe, eine Named Pipe zu verwenden, die wie eine Datei im Dateisystem prozessübergreifend angesprochen werden kann.

Dazu legt jenes Skript, dass dann auch den `ffmpeg`-Prozess startet, zuerst eine Named Pipe an. Für die Übergangsphase gegen Mitternacht ist es notwendig, mit mehreren Named Pipes zu arbeiten, deshalb bekommen sie einen eindeutigen Namen basierend auf dem Datum. Danach wird die Pipe mit File Descriptor 3 geöffnet und offen gehalten. Das ist notwendig, damit `ffmpeg` sich nicht beendet, wenn ein Bild via der Pipe fertig übermittelt wurde (siehe Listing 7.1).

```
1 today='date +%y%\%m%\%d'  
/usr/bin/mkfifo /tmp/input-$today  
3 exec 3<>/tmp/input-$today
```

Listing 7.1: Anlegen der Pipe

Hiernach kann `ffmpeg` ausgeführt werden (siehe Listing 7.2).

```
1 nice -n -15 ffmpeg -framerate 60 -f jpeg_pipe -i pipe:o -c:v libx264 -preset  
veryfast -x264-params crf=23 -c:a copy /tmp2/lapse-$today.mkv -y < /tmp/  
input-$today >> /tmp2/ffmpeg-$today.txt 2>&1 &
```

Listing 7.2: Start von `ffmpeg`

Die einzelnen Teile dieses Befehls lassen sich wie folgt erklären:

- `nice -n -15` startet `ffmpeg` mit hoher Priorität, damit sichergestellt ist, dass nicht beispielsweise eine Tätigkeit des Webservers, die auch warten kann, die Transcodierung verzögert.
- `ffmpeg` startet das eigentliche `ffmpeg` Binary.
- `-framerate 60` setzt die Bildrate auf 60 Bilder pro Sekunde.
- `-f jpeg_pipe` spezifiziert das Eingangsformat - JPEG Bilder via Named Pipe.

7. Zeitrafferaufnahmen am RPI

- `-i pipe:0` legt fest, dass via Pipe von file descriptor 0, also stdin gelesen werden soll.
- `-c:v libx264` lässt ffmpeg das Video mit der libx264 Bibliothek in H.264 komprimieren.
- `preset veryfast` wählt ein Set an Parametern² für die x264 Bibliothek (Abwägung zwischen Qualität/Dateigröße und nötiger Rechenleistung).
- `x264-params crf=23` Setzt den Constant Rate Factor, eine Abwägung zwischen Qualität und Dateigröße, 23 ist der Standardwert
- `-c:a copy` Da in diesem Schritt keine Audiodaten vorliegen wird die (leere) Tonspur übernommen
- `tmp2/lapse-$today.mkv` Legt die Datei in die das Ergebnis geschrieben wird fest.
- `-y`: Ist die Datei vorhanden, wird sie überschrieben.
- `< /tmp/input-$today` Auf diesem Weg wird via Standardinput der Inhalt der pipe eingelesen.
- `>> /tmp2/ffmpeg-$today.txt`: Status und Fehlermeldungen werden in diese Datei geschrieben, wobei `<` und `>>` jeweils pipe-Operatoren sind.
- `2>&1` (Fehler)Meldungen die sonst via stderr ausgegeben werden, werden nach stdout umgeleitet und damit in dieselbe Datei geschrieben.
- `&`: Damit wird der Prozess im Hintergrund gestartet, das Skript läuft also parallel weiter

Als letzter Schritt zum Start wird noch die Prozess-ID (PID) dieser ffmpeg Instanz in eine Datei geschrieben, damit später einfacher auf den Prozess zugegriffen werden kann:

```
echo $! > /tmp/pid_ffmpeg-$today
```

Damit können jetzt von einem anderen Skript regelmäßig Bilder in die Pipe übergeben werden, und sie werden von ffmpeg aufgegriffen und in ein Video zusammengeführt.

Dies geschieht im selben Skript, das auch die Aufnahme auslöst, siehe Listing 7.3

```
1 if [ -p /tmp/input - 'date +%y%m%d' ]; then
2 if [ -e /proc/$(cat /tmp/pid_ffmpeg - 'date +%y%m%d')/mem ]; then
3 cat /tmp2/shot.jpg > /tmp/input - 'date +%y%m%d' &
4 fi
5 fi
```

Listing 7.3: Übergabe der Bilder via Pipe

Hier wird zuerst sichergestellt, dass überhaupt eine Named Pipe existiert und dann, dass der ffmpeg-Prozess, dessen PID zuvor nach `/tmp/pid_ffmpeg-$today` geschrieben

²Die presets sind: ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow und placebo. Eine Liste mit den jeweils gesetzten Optionen: http://dev.beandog.org/x264_preset_reference.html, abgerufen 18-04-24

7. Zeitrafferaufnahmen am RPI

wurde, auch noch existiert. Ist beides gegeben, so wird mit cat das aktuelle Bild in die Pipe übergeben. Dazu wird cat mittels & am Ende des Befehls im Hintergrund gestartet, das bedeutet, das Skript läuft weiter und wartet nicht darauf, dass cat seine Ausgabe fertig über die Pipe an ffmpeg übergeben hat.

Im Prinzip könnte das Bild direkt von raspistill via Pipe an ffmpeg übergeben werden - dann ist das Optimum, nämlich wirklich ganz ohne Zwischenspeichern auszukommen - ganz erfüllt. Ein einzelnes Bild mit knapp 2 MB abzuspeichern hat aber den Vorteil, immer wieder einen kontrollierenden Blick auf die letzte Aufnahme werfen zu können. Wird zusätzlich das vergangene Bild vorgehalten, kann, um die Robustheit weiter zu erhöhen, im Fehlerfall auf dieses zurückgegriffen werden (siehe Listing 7.4).

```
1 if (( $(echo $status | cut -d" " -f1) == 0 ) ) || (($(echo $status | cut -d" "  
   -f2) != 0) ) ; then  
   /bin/cp -rf /tmp2/shot_old.jpg /tmp2/shot.jpg  
3 if (( $(echo $status | cut -d" " -f1) == 0 ) ) ; then  
   sleep 10  
5 fi  
fi
```

Listing 7.4: Fehlerbehandlung

Auf diesem Weg wird zuerst geprüft, ob raspistill eine Laufzeit von zumindest über einer Sekunde hat. Außerdem wird der Returnwert kontrolliert. Dazu bricht raspistill mit einem Fehler ab wird ebenfalls für 10 Sekunden gewartet und dann im nächsten Schleifendurchlauf fortgesetzt. Als Ersatz wird noch einmal die letzte Aufnahme an ffmpeg weitergereicht.

Im dritten Teil der Arbeit wird analysiert, wie gut dieser Ansatz funktioniert.

Eine weitere Verbesserung könnte der Einsatz der H.265 Hardwarebeschleunigung am RPI bewirken. Der Softwaresupport ist allerdings immer noch, genauso wie die Einstellungsmöglichkeiten des Encoders, eingeschränkt. Immerhin ist es nicht mehr notwendig, ffmpeg selbst mit der Unterstützung der entsprechenden Bibliotheken zu compilieren. Ein erster simpler Versuch ein Video mittels ffmpeg und hardwarebeschleunigtem H.265 zu konvertieren schlägt allerdings fehl.

7.2. Aufnahme der Einzelbilder

Bereits zur Aufnahme eines einzelnen Bildes sind einige Aspekte zu beachten. So ist anzustreben, dass die Aufnahmen in einem regelmäßigen Takt erfolgen und keinesfalls etwa Verzögerungen entstehen die sich aufsummieren. Das gelingt am einfachsten, indem die Aufnahmen als regelmäßiger Cronjob angelegt werden. Unter Debian ist die

7. Zeitrafferaufnahmen am RPI

zeitliche Auflösung hier jedoch auf eine Minute beschränkt, und ein Befehl startet auch nicht exakt zur Sekunde null dieser Minute. Daher wird ein Skript gestartet, das selbst sicherstellt, dass die Aufnahme immer alle exakt 15 Sekunden ausgelöst wird.

Ein erster Ansatz dazu wäre, in einer Schleife immer 15 Sekunden mittels `sleep 15` zu warten, dann die Aufnahme mittels `&` im Hintergrund zu starten. Ein solcher Ansatz ist aber anfällig für sich aufsummierende, durch variierende Laufzeit schwer abschätzbare Fehler, durch die Zeit die das Anfertigen der Aufnahme selbst benötigt.

Besser funktioniert, mittels des `'time'` Befehls die genaue Laufzeit der Aufnahme zu bestimmen und dann vom 15 Sekunden Intervall abzuziehen. Damit entsteht zwar ein recht komplexes Befehlskonstrukt, die Methode funktioniert aber gut.

Eleganter ist folgender Befehl, der genau bis zur nächstfolgenden vollen 15. Sekunde wartet:

```
sleep $(echo "15 - ($(date +%S.%3N -d "5 seconds") % 15)" | bc)
```

`date +%S.%3` liefert die aktuelle Sekunde auf 3 Kommastellen genau. Daraus wird der Input für `bc` zusammengebaut: `15 - aktuelle Sekunde % 15`, wobei `%` für Modulo, also den Rest einer Division mit in diesem Fall 15 steht.

Dann folgt der eigentliche Befehl, `raspistill`, der allerdings in einige weitere Befehle eingebettet ist (Listing 7.5)

```
status=$((time timeout -k 13 13 /usr/bin/raspistill --thumb none -ex night -a
  1028 -ae 15,0x00,0x8080FF -a "%Y-%m-%d %X" -w 1920 -h 1080 -n -o /tmp2/
  shot.jpg -e jpg -q 100 >> /tmp2/err_raspistill.txt 2>&1) |& grep real |
  cut -dm -f2 | cut -d. -f1 ; echo ${PIPESTATUS[0]})!
```

Listing 7.5: Aufnahme eines Fotos mit `raspistill`

Den Kern bildet der Befehl in Listing 7.6

```
1 raspistill --thumb none -ex night -a 1028 -ae 15,0x00,0x8080FF -a "%Y-%m-%d %
  X" -w 1920 -h 1080 -n -o /tmp2/shot.jpg -e jpg -q 100
```

Listing 7.6: Aufnahme eines Fotos mit `raspistill` - Kernbefehl

- `--thumb none`: Es wird kein Thumbnail erzeugt, das spart minimal Rechenleistung und Speicher.
- `-ex night`: Der Nachtmodus sorgt in der Nacht für hohe ISO Werte und lange Belichtungszeiten.
- `-a 1028 -ae 15,0x00,0x8080FF -a "%Y-%m-%d %X"`: Damit wird die Uhrzeit am oberen Bildrand eingeblendet.
- `-w 1920 -h 1080 -n -o /tmp2/shot.jpg -e jpg -q 100`: Legt Größe des Bildes, Speicherort, Bildformat und Qualitätsstufe fest.

7. Zeitrafferaufnahmen am RPI

- Mit `>> /tmp2/err_raspistill.txt 2>&1:` werden `stderr` und `stdout` in die genannte Datei umgeleitet.

Vorgelagert ist `time timeout -k 13 13` wodurch der folgende Prozess nach 13 Sekunden abgebrochen wird. Er wird dann mit einem Fehlercode beendet. Zusätzlich gibt `'time'` die tatsächliche Laufzeit aus (Listing 7.7).

```
1 |& grep real | cut -dm -f2 | cut -d. -f1 ; echo ${PIPESTATUS[0]}
```

Listing 7.7: Abfrage PIPESTATUS

Mit diesen Befehlen werden die Rückgabewerte von `raspistill` und `time` ausgelesen und in der Variable `status` abgelegt. Beispielsweise als `"0 127"`, wenn `raspistill` bereits eine Aufnahme macht und daher abbricht. Die erste Zahl bedeutet, dass der Befehl weniger als eine Sekunde gelaufen ist, was klar auf einen Fehler hinweist da die normale Laufzeit für die Aufnahme und das Abspeichern eines Bildes je nach Belichtungsdauer mindestens 5 Sekunden benötigt. Die zweite Zahl sollte, wenn `raspistill` ohne Fehler beendet wird, immer 0 sein. Wenn also die erste Zahl 0 und die zweite Zahl ungleich 0 ist, ist von einem Fehler auszugehen.

Fehler werden danach im Skript abgefangen (Listing 7.8).

```
1 if (( $(echo $status | cut -d" " -f1) == 0 )) || (($(echo $status | cut -d" "
   -f2) != 0)) ; then
   /bin/cp -rf /tmp2/shot_old.jpg /tmp2/shot.jpg
3 if (( $(echo $status | cut -d" " -f1) == 0 )) ; then
   sleep 10
5 fi
   fi
```

Listing 7.8: Fehlerbehandlung

Sollte also ein Fehler aufgetreten sein, so wird jedenfalls das zuletzt aufgenommene Bild noch einmal verwendet. In der fertigen Aufnahme zeichnen sich Fehler daher als Bildwiederholungen ab, was aber den Zeitraffereffekt nicht merklich stört. Sollte `raspistill` sofort, also in unter einer Sekunde, abgebrochen haben, wird für 10 Sekunden gewartet bis die Schleife wieder fortgesetzt wird.

7.3. Zeitraffertonaufnahmen

Wie im ersten Teil erklärt, soll die Zeitraffertonaufnahme aus einzelnen, kurzen Aufnahmen zusammengesetzt werden. Eine einzelne Aufnahme kann mit folgendem Befehl mit `SoX` erfolgen (Listing 7.9).

7. Zeitrafferaufnahmen am RPI

```
%AUDIODEV=plughw:1 rec -b 32 -r 48k -p trim 0 3 | sox - -b 16 /tmp2/a.wav
```

Listing 7.9: Tonaufnahme

Mit `AUDIODEV=plughw:1` werden die beiden I2S-Mikrofone als Aufnahmegerät ausgewählt. `rec` ist ein zu SoX gehörender Befehl zur Aufnahme. Der Treiber liefert Ton mit 48 kHz Abtastrate und 32 Bit Bittiefe (obwohl die Mikrofone nur eine Genauigkeit von 18 Bit haben, die als 24 Bit ausgegeben wird - die restlichen bits werden mit Null aufgefüllt). Mit `-p` kann das Ergebnis der mit `trim 0 3` auf 3 Sekunden begrenzten Aufnahme aus `stdout` ausgegeben werden. Auch der zweite Schritt - die Konvertierung auf 16 Bit Bittiefe - erfolgt mit SoX. Dann werden der Reihe nach einige Filter angewendet (Listing 7.10)

```
sox /tmp2/a/a.wav -p highpass 10 | sox - /tmp2/a/b.wav trim 0.5 0.7  
sox /tmp2/a/b.wav /tmp2/a/rec_$(sox /tmp2/a/b.wav -n stat 2>&1 | grep "RMS  
amplitude:" | cut -d":" -f2 | tr -d ' ' | tr -d '.').wav vol 26 dB
```

Listing 7.10: Filtern des Audiosignals

Folgende Filter werden der Reihe nach angewandt:

- `highpass 10`: Mit einem Hochpassfilter bei 10 Hz wird ein Gleichanteil aus dem Signal entfernt. Dieser Anteil ist im Signal deutlich erkennbar. Da er einfach mittels Filter entfernt werden kann wurde die Quelle nicht weiter nachvollzogen.
- `trim`: Aus dem Signal wird der Anfang, mit einem Einschwingen des Hochpassfilters, abgeschnitten.
- `vol 26 dB`: Die Mikrofone arbeiten mit sehr geringen Störgeräuschen, auch, weil die Übertragung zum RPI bereits digital erfolgt.³ Daher kann an dieser Stelle die Lautstärke angehoben werden.

In Abbildung 7.1 ist das Audiosignal nach in den einzelnen Schritten angewandten Filtern (mit anderen Werten zur besseren Darstellung) dargestellt. Deutlich zu erkennen ist die transiente Phase des Einschwingvorgangs am Beginn (bis etwa 150 ms, rot), sowie der Gleichanteil (ab etwa 300 ms konstant, rot). Mit dem Hochpassfilter wird der Gleichanteil nach einer kurzen Einschwingphase ab etwa 300 ms gänzlich entfernt, dass Frequenzen unter 10 Hz fehlen stellt keinen (hörbaren) Verlust dar.

Um, wie im ersten Teil der Arbeit vorgeschlagen, die einzelnen Aufnahmen sauber, also etwa mit einem fade-in und fade-out aneinander zu reihen, kann in SoX der Befehl `'splice'` verwendet werden. Anders als ein simples Aneinanderreihen der Audiosignale verwendet der Effekt ein Überblenden (`cross-fade`, also genau das beschriebene fade-in

³Würde die Übertragung von der Spule des Mikrofons bis zu einem Analog-Digital-Wandler am RPI analog über eine (ungeschirmte) Leitung erfolgen, so könnten allerlei Störungen eingefangen werden, etwa das bekannte von Mobiltelefonen ausgelöste Klicken.

7. Zeitrafferaufnahmen am RPI

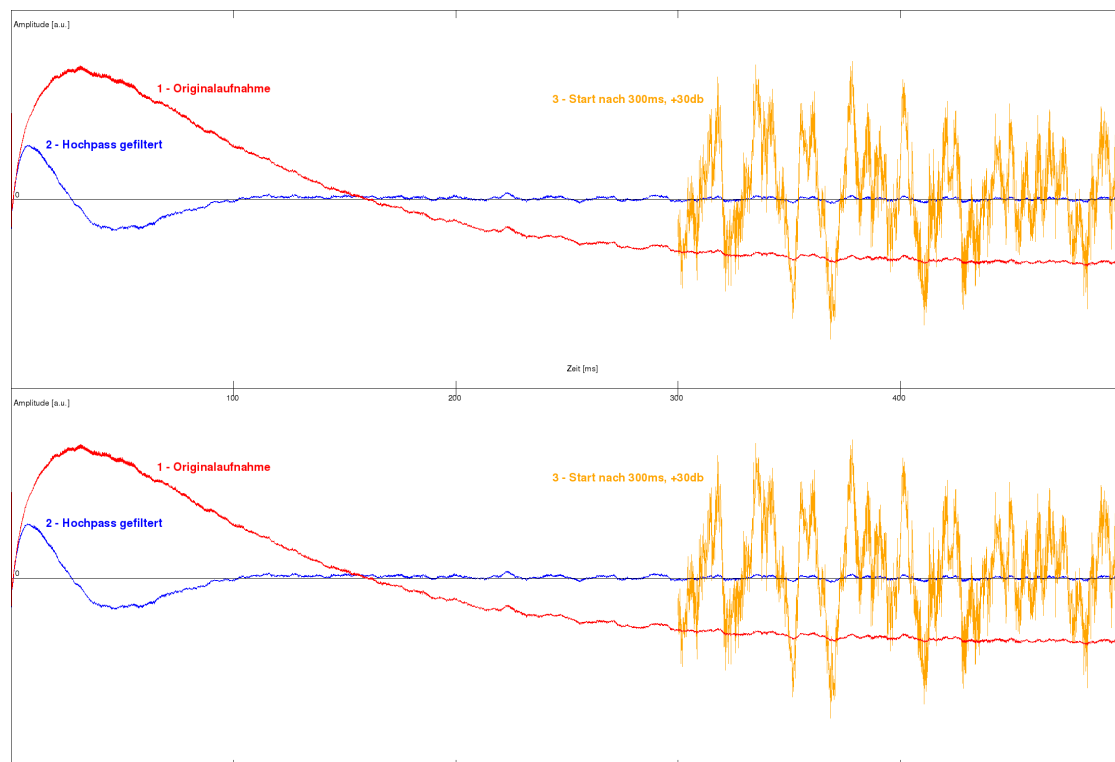


Abbildung 7.1.: Waveform-Darstellung der ersten 500 ms einer Aufnahme (rot), sowie des Audiosignals nach Anwendung des Hochpassfilters (blau) und nach Abschneiden der ersten 300 ms und Anhebung der Lautstärke (orange).

7. Zeitrafferaufnahmen am RPI

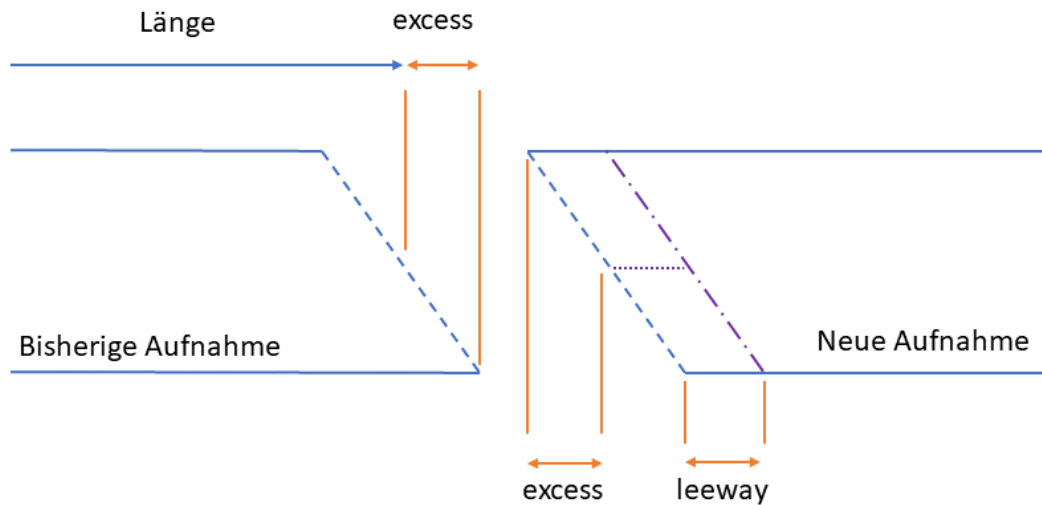


Abbildung 7.2.: Illustration des splice Befehls von SoX, eigene Grafik

und fade -out), und führt einen Vergleich der Signale durch, um sie am idealen Punkt zusammenzufügen. Damit ist auch dieser Aspekt der im ersten Teil beschriebenen WSOLA Technik hier umgesetzt. Wie bei beim Zusammenkleben von analogem Film sollte das Ende des zu spleißenden Abschnitts nach dem idealen Verbindungspunkt mit einem kleinen Überschuss (excess) abgeschnitten werden. Der Anfang des zu spleißenden nächsten Audioabschnitts sollte mit dem gleichen Überschuss (vor dem idealen Verbindungspunkt) mit einem zusätzlichen Spielraum (leeway) getrimmt werden, das ist der Bereich, in dem nach der idealen Stelle gesucht wird. Abbildung 7.2 verwendet die Analogie zu analogem Tonband zur Veranschaulichung.

Der Befehl lautet schließlich (Listing 7.11)

```
sox bisherige_aufnahme.wav neue_aufnahme.wav /tmp2/ergebnis.wav splice -q $(  
soxi -D bisherige_aufnahme.wav) ,0.2
```

Listing 7.11: Splice mit sox

Dem Befehl wird die Länge der bisherigen Aufnahme (ein eigener Befehl soxi -D liest die Länge einer Audiodatei aus) als Punkt, an dem der Spleiß erfolgen soll, übergeben und die Angabe des excess mit 200 ms - bewusst wesentlich länger als der Standardwert von 5 ms, der für den leeway beibehalten wird: Damit entstehen fließende Übergänge zwischen den einzelnen kurzen Aufnahmen.

Um die Robustheit der Abstimmung zwischen Audio- und Videolänge zu erhöhen, wird die Audiodatei, bevor sie verlustfrei mittels FLAC komprimiert abgespeichert wird, noch auf die Länge des bisher aufgenommenen Videos gekürzt. Es gilt gemäß 2.1 $l = n/r$. Die Anzahl an bisher von ffmpeg verarbeiteten Bilder wird aus der Logdatei

7. Zeitrafferaufnahmen am RPI

ausgelesen und dann durch die Bildrate von 60 dividiert, wie im folgenden Befehl dargestellt (Listing 7.12)

```
1 sox /tmp2/a.wav -C o /tmp2/'date +%y%m%d'.flac trim o $(echo $(sed -e 's/\r/\r\n/g' /tmp2/ffmpeg-${date +%y%m%d}.txt | tail -1 | cut -d= -f 2 | cut -df -f1) /60 | bc -l)
```

Listing 7.12: Zurechtschneiden der Tonspur

Das ist kein besonders eleganter oder robuster Weg, da nicht sichergestellt ist, dass ffmpeg in kommenden Versionen dieses Format als Ausgabe beibehält. Außerdem kann der Eintrag in der letzten Zeile auch eine Fehlermeldung sein, die dann an dieser Stelle zu einer falsch ausgelesenen Länge führt. Die Methode funktioniert aber dennoch gut.

Bei 60 Bildern pro Sekunde und Aufnahmen alle 15 Sekunden entstehen am Tag 96 Sekunden Video. Alle 3 Minuten wird die Tonspur dementsprechend um 200 ms verlängert um auf diese 96 Sekunden zu kommen. Aufgenommen wird 1 Sekunde, da nach dem Filtern die ersten 500 ms abgeschnitten werden und für das cross-fade zusätzlich 200 ms benötigt werden. Damit bleiben zusätzlich 100 ms, die die Tonspur im Fehlerfall alle 3 Minuten gegenüber der Videospur aufholen kann und die ansonsten abgeschnitten werden. Damit ist noch offen, zu welchem Zeitpunkt innerhalb dieser 3 Minuten die Aufnahme erfolgen soll.

In diesem Fall ist Regelmäßigkeit zu vermeiden, da z. B. mit Kirchenglocken im Stadtbereich ein regelmäßiges Tonereignis vorliegt, welches in der Zeitraffertonaufnahme sonst überproportional häufig oder gar nicht abgebildet werden würde. Eine gute Variante ist also, jeweils einen zufälligen Zeitpunkt innerhalb des Intervalls zu nutzen.

Der bessere und im Projekt auch umgesetzte Ansatz ist mit hoher Frequenz, beispielsweise alle 3 Sekunden eine Aufnahme zu machen und dann eine gute Aufnahme zu wählen. Damit liegen 60 Aufnahmen vor, aus denen ausgewählt werden kann: Je nach Motiv beispielsweise jenes Sample mit der höchsten oder der niedrigsten mittleren Lautstärke. Im vorliegenden Fall wurde ersteres gewählt.

7.4. Erstellung des Videos

Als Ende der Aufnahme wird jeweils 24 Uhr gewählt. Damit kann zwischen den Aufnahmen einfach über das Datum unterschieden werden. Am Ende eines Tages läuft ein ffmpeg-Prozess und hat - im Idealfall - ein Video aus 5760 Bildern erzeugt. Außerdem liegt eine Audiodatei im FLAC-Format mit einer Länge von genau 96 Sekunden vor. Die beiden Dateien können mit einem einfachen Befehl mit ffmpeg, ohne

7. Zeitrafferaufnahmen am RPI

dass eine Transkodierung des Videos erfolgt, mit einer als OGG Vorbis komprimierten Tonspur in einen Container zusammengeführt werden (Listing 7.13)

```
1 ffmpeg -y -i /tmp2/$yesterday.flac -i /tmp2/lapse-$yesterday.mkv -map 0:a -  
  map 1:v -filter_complex " [0:0] apad " -shortest -c:v copy -c:a libvorbis  
  -qscale:a 8 /var/www/$yesterday-audio.mkv
```

Listing 7.13: Zusammenführen von Video- und Tonspur

Mit den map-Befehlen wird eingestellt, dass im Falle einer zu langen Audiodatei diese gekürzt wird, das Video also die Gesamtlänge bestimmt.

Als erstaunlich herausfordernd stellt sich das Beenden des ffmpeg-Prozesses dar. Nachdem nur etwa immer nach 260 empfangenen Bildern die Videodaten in die Datei geschrieben werden, sollte ffmpeg nicht gezwungen (also etwa mit Signal 9) beendet werden. Mit Signal 15 hingegen kann ffmpeg eigentlich dazu gebracht werden, die verbleibenden Daten zu schreiben und sich zu beenden. Leider zeigten erste Tests, dass das nicht immer zuverlässig funktioniert, ffmpeg reagiert auf den Befehl aus unbekanntem gründen nicht zuverlässig. Als Alternative, um auch hier Robustheit sicherzustellen, werden daher weiter Bilder auch an den alten Prozess geliefert, bis dieser ein weiteres Mal Daten schreibt. Anschließend muss beim Erstellen des Videos auf 96 Sekunden gekürzt werden (Listing 7.14).

```
1 yesterday='date +%y%\%m\%d -d "yesterday" '  
  today='date +%y%\%m\%d '  
3 stat -c "%s" /tmp2/lapse-$yesterday.mkv > /tmp2/lastsize.txt  
  
5 # start new ffmpeg  
  /usr/bin/mkfifo /tmp/input-$today  
7 exec 3<>/tmp/input-$today  
  /usr/bin/nice -n -17 ffmpeg -hide_banner -framerate 60 -f jpeg_pipe -i pipe:0  
  -c:v libx264 -pix_fmt yuvj420p -preset superfast -x264-params crf=23 -c:  
  a copy /tmp2/lapse-$today.mkv -y < /tmp/input-$today >> /tmp2/ffmpeg-  
  $today.txt 2>&1 &  
9 echo $! > /tmp/pid_ffmpeg-$today  
  
11 while [[ $(cat /tmp2/lastsize.txt) = $(stat -c "%s" /tmp2/lapse-$yesterday.  
  mkv 2> /dev/null) ]]  
  do  
13 date  
  echo "sleep 1 min"  
15 sleep 60  
  ((c++)) && ((c==30)) && break  
17 done  
  
19 kill -15 $(cat /tmp/pid_ffmpeg-$yesterday)  
  sleep 60  
21 echo "kill 15"
```

7. Zeitrafferaufnahmen am RPI

```
23 kill -2 $(cat /tmp/pid_ffmpeg-$yesterday)
sleep 60
25 echo "kill 2"

27 kill -9 $(cat /tmp/pid_ffmpeg-$yesterday)
sleep 60
29 echo "kill 9"

31 #combine video and audio, move files
ffmpeg -y -i /tmp2/$yesterday.flac -i /tmp2/lapse-$yesterday.mkv -map o:a -
    map 1:v -filter_complex " [o:o] apad " -shortest -c:v copy -c:a libvorbis
    -qscale:a 8 -t 96 /var/www/$yesterday-audio.mkv
33 mv /tmp2/lapse-$yesterday.mkv /var/www/$yesterday.mkv
mv /tmp2/ffmpeg-$yesterday.txt /var/www/ffmpeg-$yesterday.txt
35 mv /tmp2/$yesterday.flac /var/www/$yesterday.flac
rm -f /tmp/input-$yesterday
```

Listing 7.14: Start von ffmpeg, Audio und Video in Container

Unter /tmp/pid_ffmpeg wird die PID von ffmpeg abgelegt. An anderer Stelle, dort wo in die Pipe geschrieben wird, sieht der entsprechende Code wie folgt aus: (Listing 7.15]

```
if [ -p /tmp/input-'date +%y%m%d -d "yesterday"' ]; then
2 if [[ $(cat /tmp2/lastsize.txt) = $(stat -c "%s" /tmp2/lapse-'date +%y%m%d
    -d "yesterday" '.mkv 2> /dev/null) ]]; then
cat /tmp2/shot.jpg > /tmp/input-'date +%y%m%d -d "yesterday"' &
4 date
echo 'piped to yesterdays ffmpeg'
6 fi
fi
```

Listing 7.15: Pipe zu ffmpeg

Das Beenden des ffmpeg-Prozesses des Vortages und der Start eines neuen ffmpeg Prozesses jeweils um 0 Uhr erfolgen, passiert beides im selben Skript, das um Mitternacht via Cronjob gestartet wird. Ein ähnliches Skript wird im Fall eines Neustarts automatisch durch Cron ausgeführt und startet den ffmpeg-Prozess.

7.5. Logging

In diesem Kapitel werden einfache Möglichkeiten, um den RPI und seine Funktion zu überwachen beschrieben. Dazu gibt es für Linux umfangreiche Softwarepakete, da die Überwachung von Systemen gerade im Serverbereich, wo Linux häufig eingesetzt wird, eine wichtige Aufgabe ist. Auf einem System wie dem RPI Zero würde Software wie

7. Zeitrafferaufnahmen am RPI

beispielsweise Munin⁴ aber zu viele Ressourcen verbrauchen. Darum soll eine einfache Textdatei als Log eingesetzt werden.

Der Hersteller gibt für seine USV genau an, wie mittels I2C die beiden Werte für Batteriespannung und Ladestand auszulesen sind, und mit welchen Faktoren multipliziert werden muss, damit sie in der richtigen Skalierung ausgegeben werden. Außerdem muss die Bytereihenfolge geändert werden, das erfolgt in einem eigenen Skript (Listing 7.16)

```
1 #!/bin/bash
2 # check stdin
3 if [ -t 0 ]; then exit; fi
4 v='cat /dev/stdin | cut -c 3-'
5 i=${#v}
6 while [ $i -gt 0 ]
7 do
8     i=$((i-2))
9     echo -n "${v:$i:2}"
10 done
11 echo
12
13 #!/bin/bash
14 echo -n "bat voltage: "
15 echo "scale=2; $(printf "%d\n" 0x$(i2cget -f -y 1 0x36 2 w | ./endian.sh)) *
16     78.125 / 1000000" | bc -l
17 echo -n "bat percent: "
18 echo "scale=2; $(printf "%d\n" 0x$(i2cget -f -y 1 0x36 4 w | ./endian.sh))
19     /256 " | bc -l
```

Listing 7.16: Auslesen der USV

Das Log selbst wird mit Hilfe des folgenden Skripts erstellt (Listing 7.17).

```
1 #!/bin/bash
2 date "+%y-%m-%d %H:%M:%S"
3 echo -n "load avg: "
4 head -c 15 /proc/loadavg
5 echo ""
6 echo -n "cpu temp: "
7 /opt/vc/bin/vcgencmd measure_temp | cut -d'=' -f 2
8 /root/ups.sh
9 echo -n "temp size: "
10 du -h /tmp2/ | tail -1 | cut -d'/' -f 1
11 echo -n "ffmpeg: "
12 ps=$(ps -eo pid,%mem,%cpu,args | grep ffmpeg | grep -v grep | grep -v timeout
13     | tr '\n' ' ')
14 echo "${ps// -hide-banner -framerate 60 -f jpeg_pipe -i pipe:0 -c:v libx264 -
15     pix_fmt yuvj420p -preset veryfast -x264-params crf=23 -c:a copy}"
```

⁴siehe <http://munin-monitoring.org>, aufgerufen 18-04-24

7. Zeitrafferaufnahmen am RPI

```
15 echo -n "free mem:  "
free -h | grep Mem | tail -c 5
17 echo -n "free swap:  "
free -h | grep Swap | tail -c 5
echo " "
```

Listing 7.17: Logging

Ein Logeintrag sieht dann wie im folgenden Beispiel aus:

```
18-04-11      22:30:27
load avg:    0.82 1.02 1.15
cpu temp:    34.7'C
bat voltage: 4.16
bat percent: 97.02
temp size:   82M
ffmpeg:      8554 37.1 21.1 ffmpeg /tmp2/lapse-180411.mkv -y
free mem:    98M
free swap:   996M
```

Das gibt einen guten Überblick über die aktuelle Auslastung von CPU (load averages, Temperatur), Arbeitsspeicher (Größe der temporär abgelegten Audio- und Videodateien, freier Arbeitsspeicher und freier Speicher in der Auslagerungsdatei), sowie den Status der USV und einen Eintrag zu ffmpeg mit der PID sowie der Speicher- und CPU-Nutzung in Prozent.

7.6. Erweiterungen

In diesem Kapitel werden, ohne alle Details zu beschreiben, da dies den Rahmen dieser Arbeit sprengen würde, einige Erweiterungen gezeigt, die nur mehr teilweise mit den Möglichkeiten des RPI Zero umgesetzt werden könnten. Daher wird für diese Versuche der RPI 2 mit 4 CPU Kernen und 1 GB Ram eingesetzt.

Wetterstation: Am RPI können Sensoren nicht nur via USB, sondern über viele andere gängige Standards, etwa I2C, aber auch über die digitalen In- und Outputs angesprochen und ausgelesen werden. Entsprechende Sensoren können sehr günstig (wenige Euro) erworben werden, und bieten sich an, um aus einem RPI eine Wetter- oder Messstation zu bauen. Das Zeitraffervideo bietet sich an, um die aktuellen Messwerte einzublenden und so den Verlauf über den Tag darzustellen.

7. Zeitrafferaufnahmen am RPI

Am RPI 2 wurden dazu mehrere Temperatursensoren (Innen- und Außenmessung) sowie Sensoren für Luftfeuchtigkeit und Luftdruck angeschlossen. Zur Stromversorgung wurde ein 30W Solarpanel eingesetzt. Die Leistungsmessung erfolgte ebenfalls über einen Strom und Spannungssensor am RPI 2 und kann als indirekte Messung der Sonneneinstrahlung verwendet werden. Außerdem ist mit entsprechend großer Pufferkapazität (in diesen Versuchen mittels gebrauchten USV Bleigelakkus) jedenfalls in den Sommermonaten ein energieautarker Betrieb möglich.

Nicht alle Sensoren können ohne weitere Maßnahmen im Freien eingesetzt werden, wenn brauchbare Messergebnisse erzielt werden sollen. Ein Temperatursensor beispielsweise sollte für eine korrekte Messung der Lufttemperatur nicht direkter Sonneneinstrahlung ausgesetzt sein, auch die anderen Sensoren sollten vor der Sonne geschützt werden. Sowohl für Luftfeuchtigkeit- und Temperatursensor ist ein ausreichend hoher Luftstrom günstig, um nicht nur einen kleinen Bereich stehender Luft zu messen. Daher werden beide Sensoren in einem kurzen, mit Alufolie ausgekleideten Rohr angebracht, durch das ein kleiner Lüfter (direkt über den RPI 2 mit 5V versorgt) einen gleichbleibenden hohen Luftstrom zieht. Da die so entstehende Strömung eine Luftdruckmessung verzerren könnte, wird der Luftdrucksensor außerhalb angebracht. Um eine Erwärmung durch direkte Sonneneinstrahlung zu vermeiden, ist das Rohr noch einmal von einem mit Aluminiumfolie ausgekleideten äußeren Rohr umschlossen. Damit sind die Sensoren gut geschützt und ein möglichst gutes Messergebnis sichergestellt. Den Aufbau zeigt Abbildung 7.3.

Um die Messergebnisse in die Aufnahmen einbetten zu können, sind einige Bearbeitungsschritte notwendig, die mittels `imagemagick` umgesetzt werden können. Sie laufen parallel und nutzen die zusätzlichen Kerne der CPU des RPI 2.

Folgendes Skript ist für die Zusammensetzung eines einzelnen Bildes zuständig. Am RPI 2 können aufgrund der stärkeren CPU nicht nur 5 statt 4 Bilder pro Minute verarbeitet, sondern auch ein Video mit höherer Auflösung (1440p60) erstellt werden (Listing 7.18).

```
#!/bin/bash
2
/usr/bin/nice -n -10 convert -normalize -extent 2560x1440 -background black /
   tmp2/shot.jpg -gravity SouthEast \
4 /tmp2/df-day.png -geometry +0+1150 -composite -gravity SouthEast \
  /tmp2/tim-temp-w-py-day.png -geometry +0+887 -composite -gravity SouthEast \
6 /tmp2/tim-speed-sh-day.png -geometry +0+635 -composite -gravity SouthEast \
  /tmp2/tim-dht-py-day.png -geometry +0+360 -composite -gravity SouthEast \
8 /tmp2/tim-pressure-py-day.png -geometry +0-28 -composite -gravity SouthEast \
  /tmp2/tim-feinstaub-sh-day.png -geometry +0+120 -composite -gravity SouthEast
   \
10 /root/clock/${1}-${2}-${3}.png -gravity NorthEast -geometry +498+400 -composite \
  /tmp2/time-..png -gravity NorthEast -geometry +497+550 -composite \
```

7. Zeitrafferaufnahmen am RPI



Abbildung 7.3.: Fotos von äußerem und innerem Rohr mit den Sensoren und Lüfter, sowie vom RPI 2 auf gebrauchtem Gehäuse eines Netzwerkschwitch montiert. Nach 2 Jahren im Außeneinsatz etwas lädiert, aber voll funktionstüchtig.

```

12 -font URW-Palladio-L-Bold -fill '#ffffff' -pointsize 120 -gravity NorthEast -
    annotate +505+0 $1 \
-14 -font URW-Palladio-L-Bold -fill '#ffffff' -pointsize 120 -gravity NorthEast -
    annotate +505+130 $2 \
-14 -font URW-Palladio-L-Bold -fill '#ffffff' -pointsize 120 -gravity NorthEast -
    annotate +505+260 $3 \
-16 -font URW-Palladio-L-Bold -fill '#88dd88' -pointsize 70 -gravity NorthEast -
    annotate +505+700 'head -2 /tmp/temps | tail -1 | xargs printf "%04.1f" '
    \
-16 -font URW-Palladio-L-Bold -fill '#dddd33' -pointsize 70 -gravity NorthEast -
    annotate +505+800 'head -1 /tmp/temps | tail -1 | xargs printf "%04.1f" '
    \
-18 -font URW-Palladio-L-Bold -fill '#885522' -pointsize 70 -gravity NorthEast -
    annotate +505+900 'xls2csv /tmp/feinstaub 2>/dev/null | grep ",," |
    tail -n 1 | cut -d\" -f 6 | xargs printf "%04.1f" ' \
-18 -font URW-Palladio-L-Bold -fill '#cc2222' -pointsize 70 -gravity NorthEast -
    annotate +505+1000 'uptime | grep -ohe 'load average[s:][: ].*' | cut -f3
    -d\" | cut -f1 -d\",' | xargs printf "%04.1f" ' \
-20 -font URW-Palladio-L-Bold -fill '#4444cc' -pointsize 70 -gravity NorthEast -
    annotate +505+1100 'head -2 /tmp/hum | tail -1 | xargs printf "%04.1f" ' \
-20 -font URW-Palladio-L-Bold -fill '#cc33cc' -pointsize 58 -gravity NorthEast -
    annotate +505+1200 'cat /tmp/pressure | xargs printf "%04.of" ' \
-20 -font URW-Palladio-L-Bold -fill '#dddddd' -pointsize 48 -gravity NorthEast -

```

7. Zeitrafferaufnahmen am RPI

```
22  annotate +505+1280 'cat /tmp/count | xargs printf "%05.0f"' \
    -font URW-Palladio-L-Bold -fill '#ffffff' -pointsize 22 -gravity NorthEast -
    annotate +505+670 "'tail -1 /tmp2/exif.txt | cut -d' ' -f2-3'" \
    -quality 100% /tmp2/shot_composed.jpg
```

Listing 7.18: Komposition eines Bildes mit imagemagick

- Der Befehl `convert` gehört zur Software `imagemagick`.
- `-normalize` normalisiert das Foto
- `-extent 2560x1440` setzt die Größe der fertig zusammengesetzten Bilder
- `-background black /tmp2/shot.jpg` positioniert das Foto

Danach folgt die Positionierung der übrigen Grafiken. Eine analoge Uhr wird als `/root/clock/$1-$2-$3.png` eingebettet, wobei das Skript aus Listing 7.18 mit der Uhrzeit als Parameter aufgerufen wird. Die entsprechenden Grafiken der Uhr für jede mögliche Sekunde (5 Aufnahmen je Minute, also 7200 Grafiken) wurden dazu zuvor einmal generiert. Danach folgt noch die Ausgabe einzelner Messwerte, und schließlich wird das Bild als JPEG mit höchster Qualität abgespeichert.

Die eingebetteten Grafiken werden von `munin` generiert. `Munin` ist ein Programm zur Überwachung von Rechnern, damit können sehr einfach über Plugins verschiedene Sensordaten und Systemkennwerte ausgelesen und in Grafiken dargestellt werden. Auch eigene Plugins können einfach erstellt werden.

Die anderen Verarbeitungsschritte können, wie für den RPI Zero beschrieben, implementiert werden. Die Ergebnisse dieser Erweiterung sind im dritten Teil der Arbeit dargestellt.

Energy Harvesting: Auch bereits am RPI 2 getestet worden ist der Betrieb an einer Solarzelle. Hier erscheint der Aufbau mit dem RPI Zero als deutlich geeigneter. Einerseits, weil bereits eine Pufferbatterie verbaut ist, und andererseits, weil der Energiebedarf deutlich niedriger ist. Je nachdem wo das Gerät eingesetzt werden soll, sind auch andere Möglichkeiten zur Energiegewinnung denkbar, etwa über eine kleine Turbine (hier gibt es fertige Bauteile mit ausreichend Leistung bereits im einstelligen Eurobereich⁵) oder mittels Wind. In allen drei Fällen (Solar, Wasser, Wind) ist eine entsprechend ausgelegte Elektronik notwendig. Auch hier gibt es fertige Module günstig zu kaufen. Für den Betrieb des RPI 2 an einer 30W Solarzelle mit Bleiakku als Puffer wurde beispielsweise ein Maximum Power Point Tracking (MPPT) Solarmodul (siehe Abbildung 7.4) mit Batterieladesteuerung für unter 10 Euro erfolgreich getestet. Mittels MPPT wird der Betriebspunkt der Solarzelle optimiert, damit konnten durch optimale Ausrichtung des Panels bei entsprechender Sonneneinstrahlung und nicht zu hohen Temperaturen sogar Leistungen von deutlich über 30W entnommen werden. Am RPI Zero wäre

⁵z. B. hier <https://www.ebay.at/itm/381582940287>, aufgerufen 18-04-24

7. Zeitrafferaufnahmen am RPI

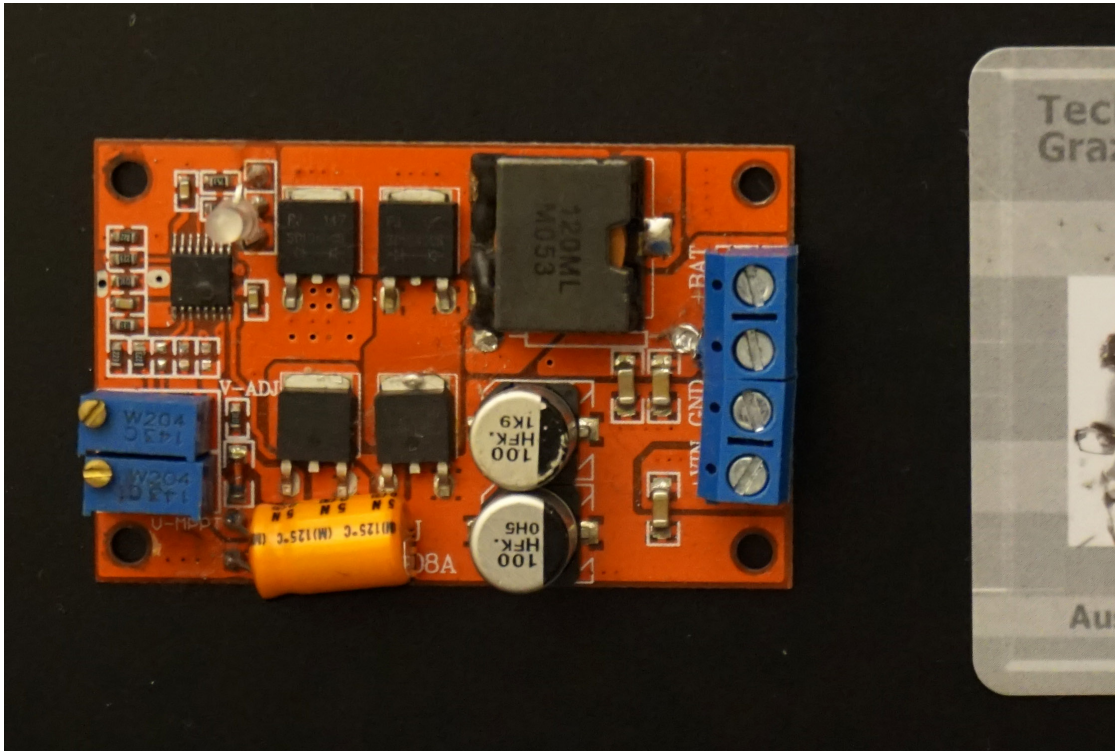


Abbildung 7.4.: MPPT-Modul

aufgrund des geringeren Energiebedarfs dann auch ein ganzjähriger Betrieb rein aus Solarenergie denkbar. Ein autarker Betrieb mittels erneuerbaren Energien beleuchtet, neben den komplexeren Aufgabenstellungen bei der Einrichtung, auch Aspekte des Umweltschutzes und der Energiegewinnung.

Feinstaub: Noch nicht umgesetzt aber eine spannende weitere Ergänzung ist, besonders in der feinstaubbelasteten Stadt Graz, der Anschluss eines PM_{2.5} und PM₁₀ Luftsensors. Obwohl die Messung via Laserbeugung-Partikelgrößenanalyse aufwändig ist (auch schon mechanisch, denn die Luft muss über einen Lüfter im Sensor am Laser vorbei bewegt werden), existieren Sensoren, die für unter 20 USD verkauft werden (siehe Abbildung 7.5⁶). In der Zwischenzeit wird der Wert der nächstgelegenen Messstation der Stadt Graz, die diese Werte als Open Data zur Verfügung stellt, herangezogen.

Mobile Montage und Kühlung: Als Erweiterung für den Aufbau mit dem RPI Zero bietet sich eine Möglichkeit an, das Gerät drehbar, auf Seilen oder Schienen zu montieren um die im ersten Teil beschriebenen Hyperlapseaufnahmen zu ermöglichen. Es gibt einfache Motorsteuerungen die auf den RPI abgestimmt sind, sodass die entsprechenden Servos ebenfalls aus Skripten heraus gesteuert werden könnten.

⁶das Modell kann z. B. unter <https://www.aliexpress.com/item/nova-PM-sensor-SDS011-High-precision-laser-pm2-5-air-quality-detection-sensor-module-Super-dust/32317805049.html> (aufgerufen 18-04-24) bezogen werden

7. Zeitrafferaufnahmen am RPI



Abbildung 7.5.: Feinstaubsensor

7. Zeitrafferaufnahmen am RPI

Für eine aktive Kühlung, die in dem Zeitraum in dem das Gerät getestet wurde noch nicht notwendig war, bietet sich eine einfache Ansteuerung über einen MOSFET Transistor an. Damit kann der Lüfter von den 5V Versorgungsspannung betrieben und über einen der GPIO Pins mittels PWM gesteuert werden: Der Lüfter sollte warme Luft vorne aus dem Gehäuse blasen, sodass dort, wo ohnehin eine Öffnung für den Stromanschluss vorgesehen ist, kalte Luft nachströmen kann. Er sollte aktiviert werden, sobald die CPU-Temperatur beispielsweise 60 Grad erreicht und immer kurz vor einer Tonaufnahme für deren Dauer in der Drehzahl (eventuell bis auf 0) reduziert werden, um Störgeräusche zu vermeiden. Der Transistor ist so zu wählen, dass er bereits bei 3.3 V voll durchschaltet. Ein entsprechender 'logic level' Transistor ist der IRLZ24NPBF⁷, mit dem bis zu 45W geschaltet werden können.

Hochkontrastbilder und Dynamikkompression: Im Rahmen von Zeitrafferaufnahmen werden auch mit Kameras, die weder nativ noch durch entsprechend hohe Aufnahmegeschwindigkeit in der Lage sind, Hochkontrastbilder (HDR) zu erzeugen, diese interessanten Aufnahmen möglich. Da für jedes Einzelbild eine Belichtungsreihe aufgenommen werden muss, also beispielsweise 3 oder 5 Fotos, müssten für ein Video mit 30 Bildern pro Sekunde in jeder Sekunde 90 oder 150 Aufnahmen entstehen. Daraus ergeben sich kurze maximale Belichtungsdauern insbesondere für das hellste Bild in der Reihe; also sehr hohe Anforderungen an die Kamera. Handelt es sich aber um eine Zeitrafferaufnahme, so ist mehr Zeit vorhanden um die Belichtungsreihen aufzunehmen und zu verarbeiten. Besonders im Zuge der anschließenden Dynamikkompression entstehen so ansprechende und außergewöhnliche Bilder - das soll auch am RPI Zero versucht werden. Daher wird an dieser Stelle als kurzer Exkurs in Hochkontrastbilder (HDR - High Dynamic Range) und Dynamikkompression (tone mapping) eingeführt.

HDR Aufnahmen sind Aufnahmen, die große Helligkeitsunterschiede wiedergeben können. Üblich sind, wie im ersten Teil der Arbeit erwähnt 8 Bit je Farbkanal, also jeweils 256 Helligkeitsstufen für Rot, Grün, und Blau. Es existieren inzwischen bereits einige Computer- und vor allem Fernsehmonitore die auch 10 Bit unterstützen und mit Ultra Blu-Rays auch Datenträger, die entsprechendes Videomaterial speichern. Abgesehen von speziellen Kameras oder mit am Computer generierten Grafiken können HDR Aufnahmen auch aus einer Belichtungsreihe von Fotos mit geringem Dynamikumfang erstellt werden. Werden diese Bilder dann wieder in eine Datei mit Standarddynamikumfang zurück überführt, handelt es sich strenggenommen nicht mehr um HDR Bilder. Dieser Vorgang, zu dem verschiedene Algorithmen die jeweils sehr unterschiedliche Ergebnisse erzielen existieren (Reinhard, 2010, S. 233ff), wird Tone-Mapping genannt und ist recht rechenintensiv.

⁷siehe http://www.produktinfo.conrad.com/datenblaetter/150000-174999/162869-da-01-en-IRLZ_24_N.pdf, aufgerufen 18-04-24

7. Zeitrafferaufnahmen am RPI



Abbildung 7.6.: Beispiele für HDR Aufnahmen

In Abbildung 7.6 sind mehrere auf diesem Weg entstandene Bilder dargestellt⁸.

Ein direkterer Weg, um aus verschiedenen belichteten Fotos ein Bild mit einem hohen Kontrastumfang zu erzeugen, ist "Exposure Fusion" (Mertens, 2007). Dabei wird keine Zwischenstufe in Form eines HDR-Bilds mit anschließendem Tonemapping generiert. Dieses Verfahren ist wesentlich weniger rechenintensiv als der Weg über HDR.

In allen Varianten, die auf Belichtungsreihen zurückgreifen, sind außerdem das automatisierte Ausrichten der Bilder (sofern sich die Kamera bewegt hat - im in dieser Arbeit betrachteten Anwendungsfall nicht relevant) und wenn sich das Motiv zwischen den Aufnahmen bewegt sogenanntes Deghosting notwendig (Tursun u. a., 2015). Dadurch wird der Rechenaufwand weiter erhöht.

⁸Von links oben im Uhrzeigersinn, erstellt jeweils mit Photomatix Pro:

- Autos aus der UDSSR, Militärmuseum „Kampfesruhm des Ural“, Werchnjaja Pyschma, Russland, 22. November 2015, 11 Fotos, Canon PowerShot SX280 HS
- Sonnenaufgang im Winter, Dominikanergründe, Graz, Österreich, 27. Jänner 2012, 7 Fotos, Canon PowerShot SX230 HS
- Herzmuscheln, Verkaufsstand am Markt, Paris, Frankreich, 13. April 2012, 9 Fotos, Canon PowerShot SX230 HS
- Weg mit Brücke im Wald, Alpbach, Österreich, 25. August 2016, 8 Fotos, Canon PowerShot SX280 HS

7. Zeitrafferaufnahmen am RPI



Abbildung 7.7.: Beispiele für Aufnahme mit Tilt-und-Shift Filter

Beide Methoden wurden erprobt, einerseits Exposure Fusion mit der Software `enfuse` als auch der Weg über HDR Bilder aus Belichtungsreihen mit anschließendem Tone-mapping mit `luminanceHDR`. Die Ergebnisse sind im dritten Teil der Arbeit dargestellt.

Tilt-und-Shift Aufnahmen: Neben HDR sind insbesondere auch Filter, die eine Aufnahme, die jener eines Tilt-und-Shift-Objektiv entsprechen erzeugen spannend: Damit entsteht visuell der Eindruck, dass eine Miniatur aufgenommen wird. Der Effekt ist einfach mit `imagemagick` zu implementieren, allerdings rechnet der RPI Zero mehrere Minuten an einem Bild - damit wird der Effekt nur für extremere Zeitraffer möglich. Auf einem RPI 2 oder 3 könnte die Operationen dank der zusätzlichen CPU-Kerne ohne relevante Verzögerungen möglich sein. Abbildung 7.7 zeigt eine solche am RPI Zero entstandene und bearbeitete Aufnahme, Listing 7.19 die Befehle.

```
convert ausgangsbild.jpg -sparse-color Barycentric '0,0 black 0,%h white' -  
function polynomial 4,-4,1 blurmap.png  
convert ausgangsbild.jpg -sigmoidal-contrast 5x20% blurmap.png -compose Blur  
-set option:compose:args 10x0 -composite blurmap.png -compose Blur -set  
option:compose:args 0x10 -composite tilt_shift_foto.jpg
```

Listing 7.19: Tilt-und-Shift Filter mit `imagemagick`, adaptiert von <http://www.imagemagick.org/Usage/photos/>, abgerufen 18-04-25

7.7. Zusammenfassung

Im zweiten Teil dieser Arbeit wurden alle Schritte gezeigt, um den Bau des beschriebenen Gerätes nachvollziehen zu können: die relevanten Einstellungen, Skripte und der Aufbau der Hardware. Außerdem wurden Beispiele für Erweiterungen getestet, die auf einem RPI 2 mit höherer Rechenleistung eingesetzt werden können. Im folgenden, dritten Teil der Arbeit wird analysiert, inwieweit die im ersten Teil der Arbeit gesteckten Ziele erfüllt werden konnten.

Teil III.

Analyse

8. Ergebnisse

Im dritten Teil dieser Arbeit werden die Ergebnisse präsentiert und gezeigt inwieweit das auf dem RPI Zero aufgebaute Gerät (im Einsatz zu sehen in Abbildung 8.1) die im ersten Teil motivierten Ziele erreichen kann:

- Automatische Erstellung eines Zeitraffervideos in FullHD Auflösung und 60 Bildern pro Sekunde (1080p60)
- Mehrere Aufnahmen pro Minute
- Aufnahme von Ton im Zeitraffer der als "natürlich" wahrgenommen wird
- Robustheit
- Weiterbetrieb bei Stromausfall
- Automatischer Start der Aufnahme nach Neustart des RPI
- Einsatz von FOSS Software
- Gesamtkosten von nicht mehr als etwa 50 Euro

Dazu wird zuerst das Motiv kurz vorgestellt, Bilder, aus denen sich das Zeitraffervideo zusammensetzt, dargestellt und Ergebnisse aus den Logdateien und von den Aufnahmen im Zeitraum von knapp zwei Monaten präsentiert.

Auch zu den im 2. Teil beschriebenen Erweiterungsmöglichkeiten werden einzelne Ergebnisse (Bilder der entstandenen Aufnahmen) präsentiert. Das eigentliche Ergebnis - die Videodateien - ist für einen Abdruck leider ungeeignet.

8.1. Das Motiv

Bei dem im Zeitraffer aufgenommenen Bauvorhaben handelt es sich um fünf bis achtgeschossige Wohnhäuser, die im Grazer Stadtteil Jakomini auf den Dominikanergründen im Besitz des Benediktinerstift Admont errichtet werden. Der Plan sieht vor, auf zwei nebeneinander liegenden Planungsgebieten in einer Tiefe von 8m Gebäude mit einer maximalen Höhe von 17 bis 26 m zu errichten. Als Baubeginn gibt die Meldung der Bauarbeiten den 19. Februar 2018 an, als Ende der Arbeiten den 30. September 2019¹.

¹aus: „Bestätigung der Meldung der Bauarbeiten“, Aushang an der Baustelle.

8. Ergebnisse



Abbildung 8.1.: Die Kamera vom der Baustelleneinfahrt aus fotografiert

In Abbildung 8.2 ist der Sichtwinkel der Kamera (und damit ihre Position) am Bauplan der Stadt Graz eingezeichnet.

8.2. Aufnahmen

Dieser Abschnitt zeigt zunächst einige Aufnahmen die mit dem RPI Zero entstanden sind (Abbildungen 8.3, 8.3, 8.4 und 8.5) dann folgen Ergebnisse der Tests mit enfused und luminanceHDR für HDR Bilder (Abbildungen 8.6 und 8.7) die einige der Erweiterungen mit dem RPI 2 darstellen (Abbildungen 8.8, 8.9, 8.10 und 8.11).

8.3. Anwendungsbeispiele

Im ersten Teil dieser Arbeit wurden bereits Beispiele genannt, welche Vorgänge durch Zeitrafferaufnahmen sichtbar werden, die mit einer normalen Videoaufnahme (vom Menschen) nicht erkennbar wären – die Beispiele spannen einen weiter Bogen von tierischem Paarungsverhalten bis zur Bewegung von Gletschern. Im vorliegenden Beispiel einer Baustelle lassen sich auch viele verschiedene Aspekte beobachten, die ohne Zeitraffer versteckt bleiben (nicht alle davon haben mit der Baustelle selbst zu tun):

- Die Bewegung der Sonne, direkt und indirekt über den Schattenwurf der Bäume.
- Das Trocknen der Zufahrtsstraße, die mehrmals von einem Kehrwagen gereinigt wird.

8. Ergebnisse

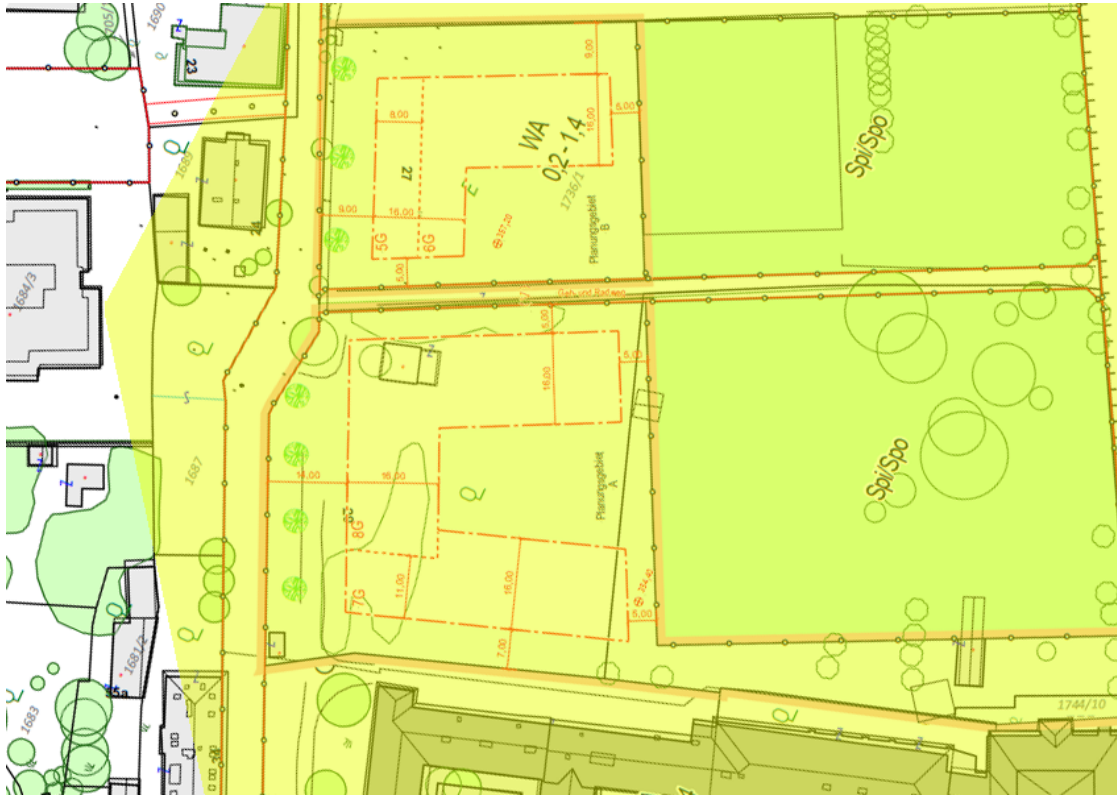


Abbildung 8.2.: Gedrehter Ausschnitt aus dem Bebauungsplan 06.23.0, Stadt Graz, 2016 (aus https://www.graz.at/cms/dokumente/10274696_7768145/a710d296/top12%2BBLGEN.pdf, abgerufen 18-04-25). Gelb eingefärbt: ist der von der Kamera erfasste Bereich.



Abbildung 8.3.: Aufnahme vom 18-04-19, 8:33:15, Betonpumpe im Einsatz

8. Ergebnisse

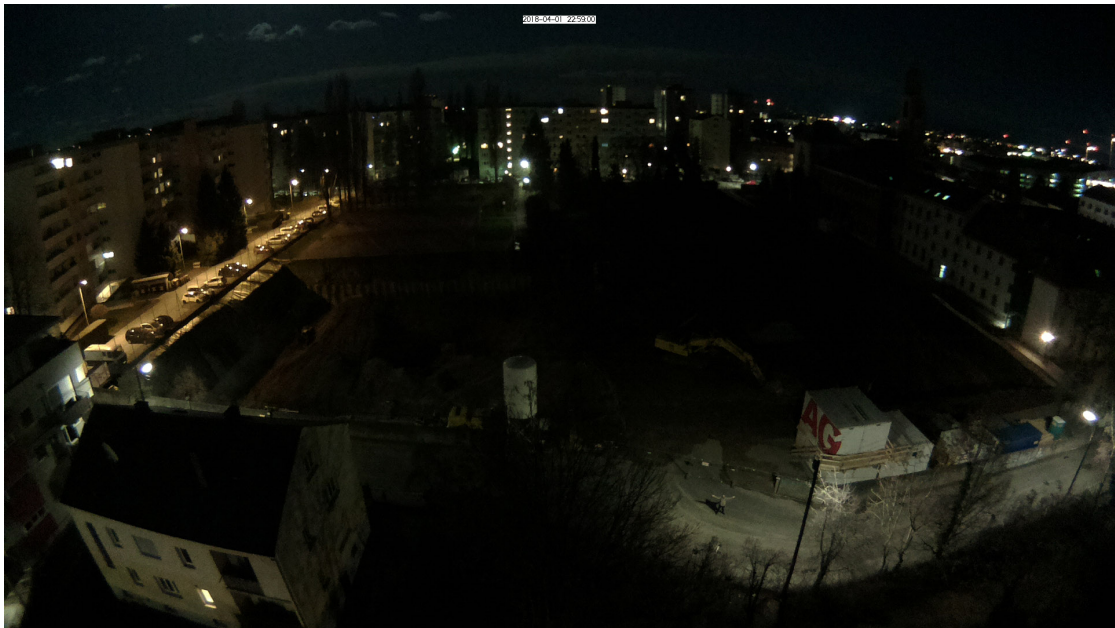


Abbildung 8.4.: Aufnahme vom 18-04-01, 22:59:00, April 1st Selfie

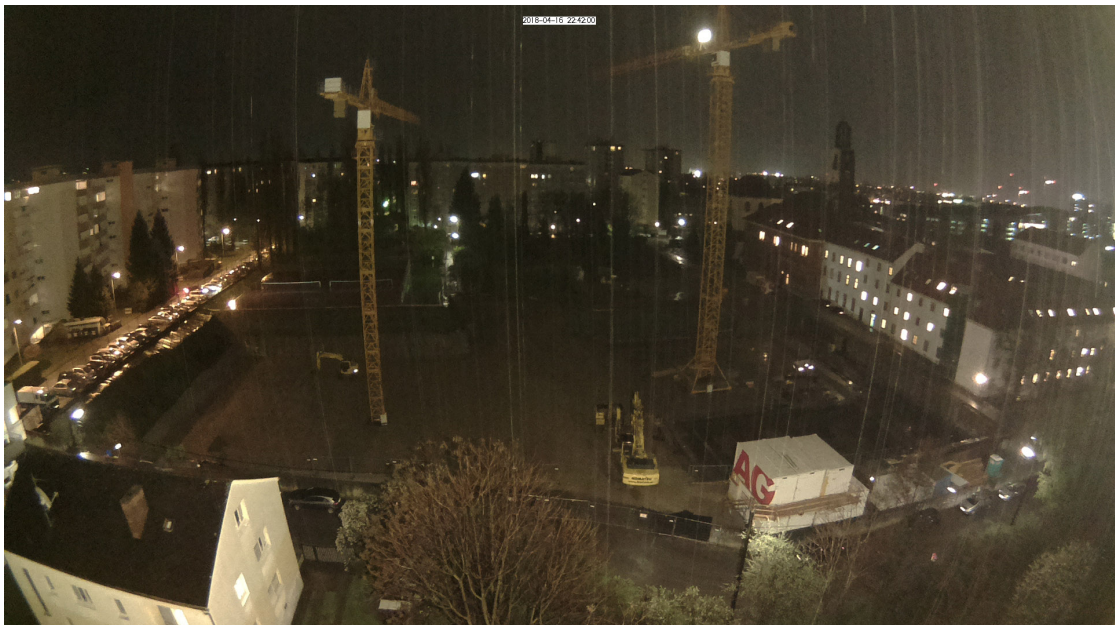


Abbildung 8.5.: Aufnahme vom 18-04-16, 22:42:00, Beleuchtete Baustelle, Regentropfen sichtbar

8. Ergebnisse



Abbildung 8.6.: Mehrere Aufnahmen vom 18-04-04, 09:55, mit enfused am RPI Zero zusammengesetzt, Ghosting deutlich zu erkennen



Abbildung 8.7.: Mehrere Aufnahmen vom 18-04-04, 10:04, mit luminanceHDR am RPI Zero zusammengesetzt, Ghosting entfernt, Algorithmus: mantiuko6

8. Ergebnisse



Abbildung 8.8.: Aufnahme vom 17-06-25, 02:03:06, mit dem RPI 2 mit Sensordaten, Blitz



Abbildung 8.9.: Aufnahme vom 17-06-25, 02:01:06, mit dem RPI 2 mit Sensordaten, von Blitz hell erleuchtete Szene

8. Ergebnisse

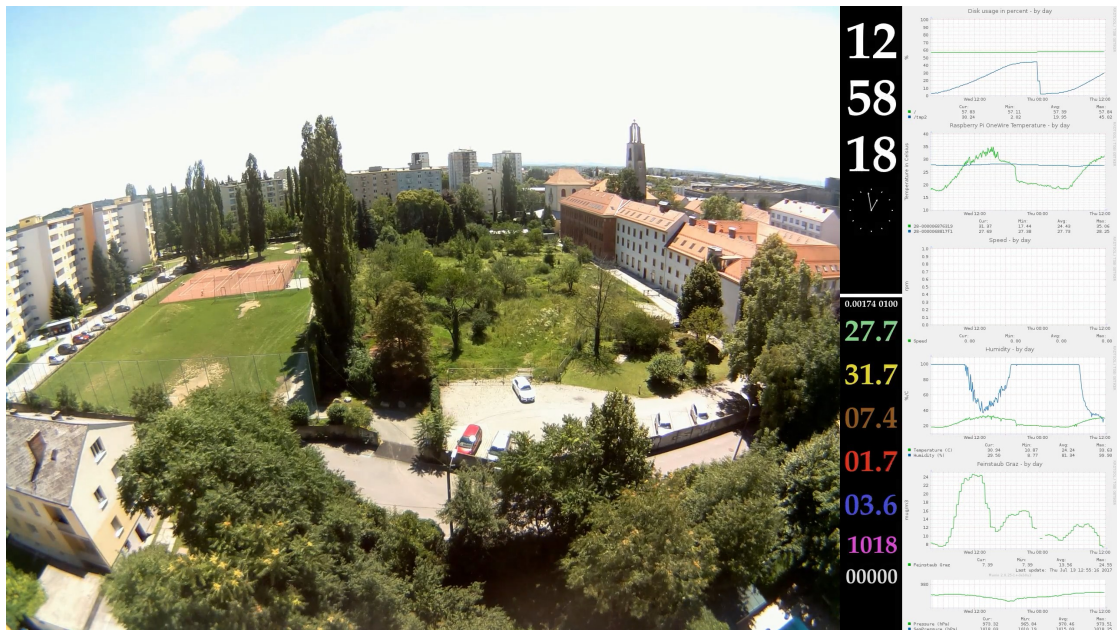


Abbildung 8.10.: Aufnahme vom 17-07-13, 12:58:18, mit dem RPI 2 mit Sensordaten, sonniger Sommertag

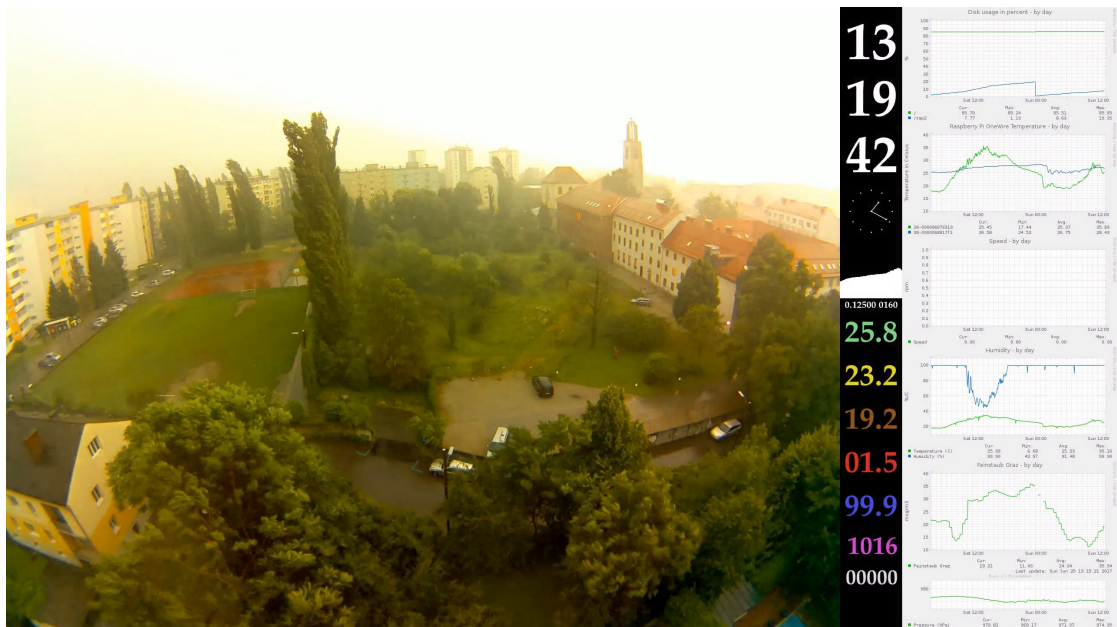


Abbildung 8.11.: Aufnahme vom 17-06-25, 13:19:42, mit dem RPI 2 mit Sensordaten, heftiger Sturm

8. Ergebnisse

- Die komplexe Baustellenlogistik, insbesondere in der frühen Bauphase die zahlreichen (etwa ein Dutzend) schweren Lastkraftwagen. Möglichkeiten, die Logistik weiter zu verbessern, etwa indem die LKWs besser getaktet werden und so Stillzeiten vermieden werden: Die LKW warten bereits ab etwa 5:30 Uhr morgens, um 6:00 beginnt der Bagger sie zu befüllen, ist damit allerdings fertig, bevor der erste LKW nach etwa 45min wieder leer zurückkommt. Dann treffen aber zugleich mehrere leere LKW ein und müssen warten. Da abwechselnd Bagger auf LKW und LKW auf Bagger warten, gibt es offensichtlich Optimierungspotential.
- Die Parallelisierung von Arbeiten, beispielsweise ist über die gesamte frühe Bauphase (Anfang März bis Ende April) hinweg ein schwerer *Komatsu* Raupenbagger im Einsatz um Erde abzutragen, gleichzeitig werden Stahlbetonpfähle gesetzt um angrenzende Gebäude und den Sportplatz zu schützen.
- Der Aufbau der beiden Turmdrehkräne am 4. und 5. April kann im Detail verfolgt werden. Anders als auf einem Einzelbild oder einem normalen Video, zeigt die Zeitrafferaufnahme wie stark sich der Turm bewegt, wenn Lasten bewegt werden.
- Ab 19. April lässt sich die Arbeit der großen, 5-achsigen, mobilen Betonpumpe beobachten.

Die Betonarbeiten ab Mitte April zeigen den großen manuellen Aufwand für das Verlegen der Stahlbewehrung der ersten Betonschicht über der Sauberkeitsschicht.

Für diese Beispiele wurde ein 25-minütiger Zusammenschnitt von Aufnahmen vom 13. März 2018 bis 25. April 2018 verwendet. Arbeitsfreie Zeiten wurden für dieses Video, das eine Beschleunigung um einen Faktor 900 zur Echtzeit darstellt, entfernt. Wird dieses Video noch einmal um einen Faktor 60 beschleunigt (insgesamt also 54.000-fache Echtzeit), wird es möglich den baulichen Strukturen beim „Wachsen“ zuzusehen.

Die im zweiten Teil der Arbeit genannten Erweiterungsmöglichkeiten in Richtung einer Wetterstation erlauben es, Grafen einzublenden, auf denen die Entwicklung von Temperatur, Luftdruck, Luftfeuchtigkeit etc. verfolgt werden können. In Abbildungen 8.8, 8.9, 8.10 und 8.11 sind solche Graphen sichtbar. Damit lassen sich Wetterumschwünge und ihre Vorzeichen in den Messwerten beobachten.

Viele spannende Beispiele, die etwa für den Biologieunterricht spannende Einblicke bringen könnten, finden sich im Internet, beispielsweise bei youtube user *Temponaut*² mit einem Fokus auf Pflanzenwachstum und Verrottung von Lebensmitteln.

²url<https://www.youtube.com/user/TEMPoNAUT/videos>, abgerufen 18-05-04

8.4. Robustheit

Der RPI Zero funktioniert in der hier gezeigten Konfiguration sehr stabil, aktuell zeigt der Befehl uptime:

```
01:37:28 up 6 days, 23:14, 5 users, load average: 0.40, 0.70, 0.63
```

Nach einem Stromausfall übernimmt die USV für zumindest 10 Stunden die Energieversorgung. Da der Ladestand ausgelesen werden kann, kann auch eine Abschaltung ausgelöst werden. Wenn das Gerät dann wieder gestartet wird, setzt die Aufnahme von Bildern und die Generierung von Videos wieder sogleich ein. Damit ist dem im ersten Teil dieser Arbeit gestellten Anspruch an Robustheit Genüge getan.

In Abbildung 8.12 sind die Auslastung, Temperatur der CPU und Auslastung des temporären Speichers für die Audio, Bild- und Videodateien über einen Zeitraum vom frühen Morgen am 21. April bis 8 Uhr am Abend des nächsten Tages aufgetragen. Deutlich ist zu erkennen, wie kurz nach Mitternacht die Videodatei aus der Ramdisk entfernt und die temporären Dateien des Vortages gelöscht werden. Am 22. April erfolgt das manuell, um die Aufnahme des Tages zu sichern, da das Gerät zu diesem Zeitpunkt für Fotoaufnahmen abgebaut und zerlegt wird. Deshalb sinkt auch am frühen Nachmittag des 22. die Temperatur um einige Grad: Zu diesem Zeitpunkt wurde der RPI Zero zur Aufnahme von Fotos für diese Arbeit aus dem Gehäuse und von der direkten Sonne entfernt. Zur besseren Einschätzung, wie sehr die Temperatur durch das Gehäuse mit wenig Luftzug steigt, zeigt Abbildung 8.13 die Außentemperatur in Graz (Messstation Universität Graz, übernommen aus https://at.wetter.com/wetter_aktuell/rueckblick/?id=ATAT20125&sid=11290, abgerufen 18-04-22).

8.5. Hardwarekosten

Tabelle 8.5 listet die verwendeten Komponenten, ihren ungefähren Preis in Euro, sowie jeweils eine mögliche Bezugsquelle. Die groben Gesamtkosten belaufen sich auf etwa 55 Euro. Gespart werden können etwa 3 Euro beim RPI Zero W (Version ohne Wlan und Bluetooth), 3 Euro bei der Kamera (ohne Weitwinkel), 13 Euro bei der USV und dem Akku, und 9 bei den Mikrofonen (sehr günstiges USB Mikrofon als Ersatz) - damit lassen sich die Kosten auf unter 30 Euro senken.

8. Ergebnisse

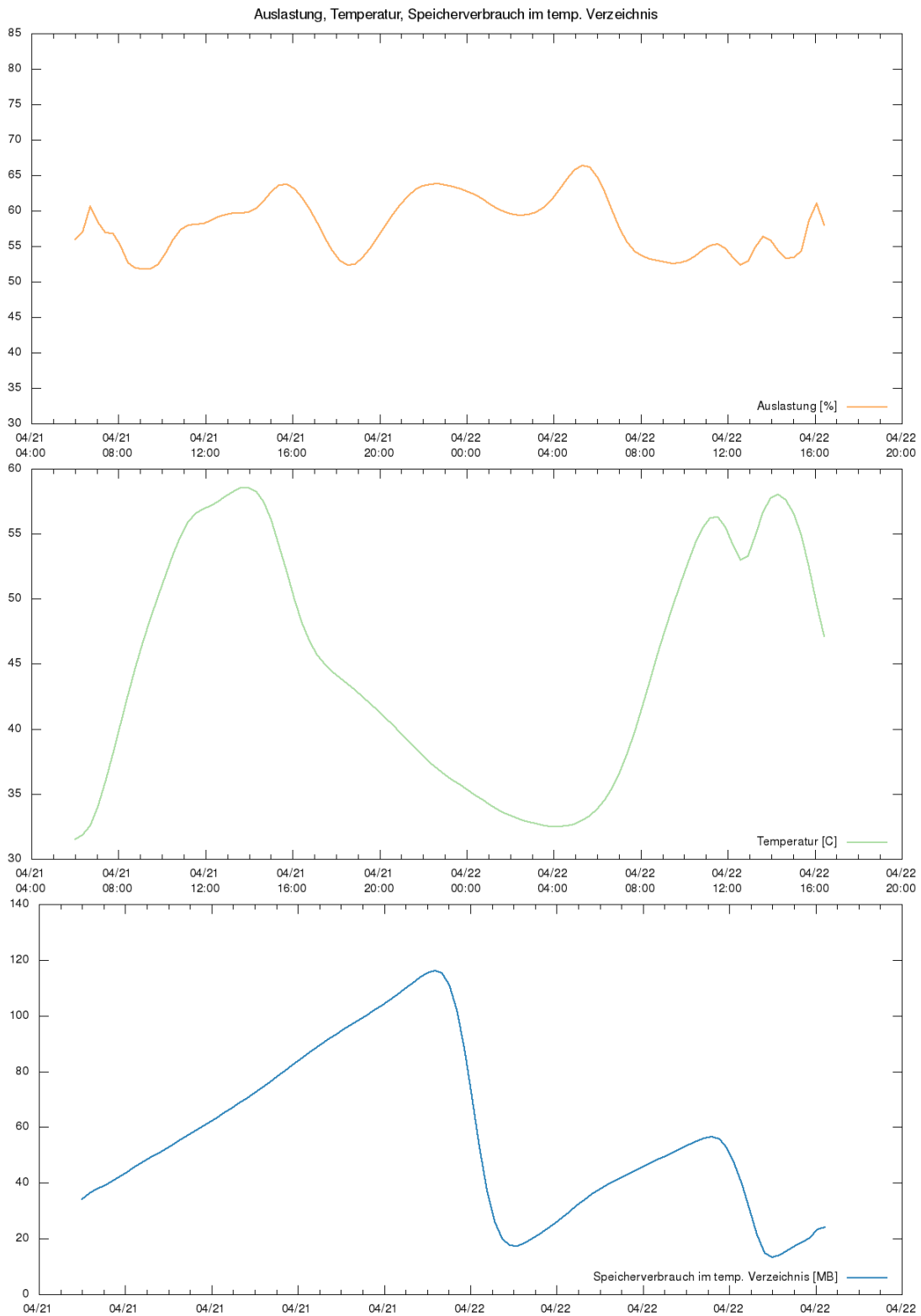


Abbildung 8.12.: Logs über Auslastung, Temperatur der CPU und Auslastung des temporären Speichers für die Audio, Bild- und Videodateien.

8. Ergebnisse

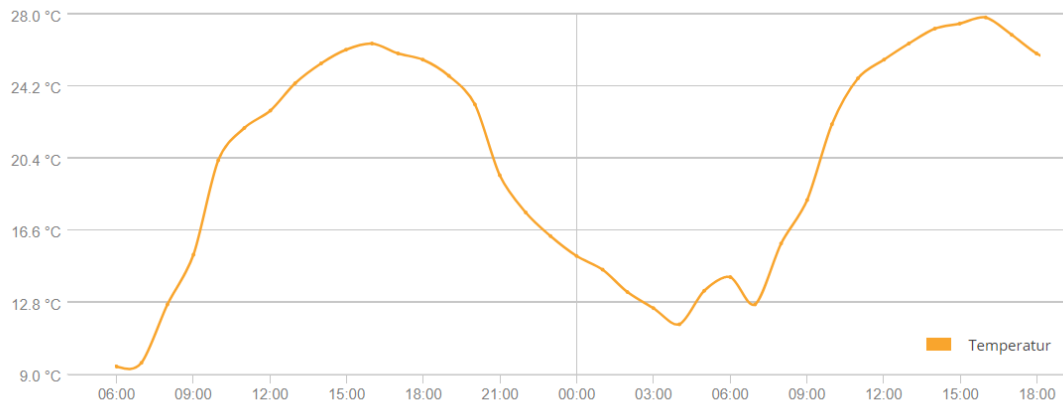


Abbildung 8.13.: Temperaturverlauf in Graz am 21. und 22. April 2018

Tabelle 8.1.: Hardware Einkaufsliste

Komponente	Preis	Beispiel für Bezugsquelle
RPI Zero W	8 Euro	www.canakit.com
RPI Kamera	18 Euro	www.ebay.at/itm/221575476514
Geekworm USV	10 Euro	www.dx.com
Akku (LiPo 18650)	3 Euro	eu.nkon.nl
Mikrofone MEMS micro	11 Euro	www.mouser.at
SD Speicherkarte	5 Euro	www.moerth-online.at

8.6. Aufwandsabschätzung für den Nachbau

Vorausgesetzt, dass alle nötigen Bauteile verfügbar sind, eine gewisse Übung im Umgang mit Linux vorliegt und keine Probleme auftreten, sollte sich folgender Zeitplan einhalten lassen:

- Zusammenbau (Mikrofone, Batterie, USV, Gehäuse etc.): 3 - 5h
- Installation von Raspbian, Einstellungen setzen, Skripte kopieren: 1h

Realistischer ist natürlich eine deutlich längere Beschäftigung, da vermutlich viele Techniken und Kenntnisse erst erworben werden müssen - Umgang mit Raspbian / Linux, BASH, aber auch Löten und ein gewisses handwerkliches Geschick wenn es um die Wahl des Gehäuses und den Einbau geht. Insbesondere sobald von den für diese Arbeit über gut 2 Jahre entwickelten Vorschlägen abgewichen wird, Ergänzungen und Variationen erprobt werden, kann die Beschäftigung mit dem Projekt auch deutlich länger dauern.

Kurzanleitung: Um das Gerät nachzubauen sind nur wenige Schritte notwendig

1. USV aufstecken und Pins anlöten
2. Akku an USV anschließen
3. Mikrofone gemäß Anleitung auf der Herstellerwebsite an den RPI anschließen³
4. Kamera mit Flachkabel über DSI anschließen
5. Betriebssystem auf SD Karte installieren und Karte einschieben
6. Konfiguration gemäß Teil 2 dieser Arbeit durchführen
7. Skripte auf die SD Karte kopieren

Zusammenfassung

In diesem dritten Teil der Arbeit wurden die Ergebnisse präsentiert und in der Analyse gezeigt, dass alle im ersten Teil gesteckten Ziele erfüllt werden konnten.

³<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-i2s-mems-microphone-breakout.pdf>, abgerufen 18-04-25

9. Zusammenfassung

Diese Arbeit liefert eine Anleitung um aus einem Raspberry Pi Zero und weiterer Hardware ein Gerät zu bauen, das robust Zeitrafferaufnahmen mit Bild und Ton erstellt. Am Ende jedes Tages liegt eine H.264 komprimierte Videodatei mit Ogg Vorbis komprimierter Tonspur vor, die 5760 Einzelbilder enthält und 96 Sekunden lang ist.

Die im ersten Teil der Arbeit motivierten Anforderungen konnten allesamt erfüllt werden. Das Projekt ist mit einem Budget von 50 Euro, mit Abschlägen auch deutlich günstiger, umsetzbar und kann innerhalb einiger Stunden nachgebaut werden. An vielen Stellen bietet es sich aber an, eigene Variationen zu erproben - dazu macht die Arbeit diverse Vorschläge, von denen einige auch getestet wurden. Das Projekt motiviert damit eine interdisziplinäre Auseinandersetzung mit künstlerischen und technischen Aspekten von Fotografie, Bild-, Ton-, und Videotechnologien, mit Elektrotechnik, Informatik bis hin zu rechtlichen Fragen. Das Projekt erscheint damit als Basis für Schulprojekte geeignet, um Schüler/innen mit unterschiedlichen Interessen anzusprechen und zur Beschäftigung mit Themen der Informatik und Elektrotechnik zu bewegen und für ein technisches Studium zu gewinnen. Da die Arbeit und alle entstandene Software unter einer freien Lizenz stehen, ist es auch für Lehrer/innen einfach möglich, die Inhalte und Ansätze für eigene Projekte zu benutzen.

Glossar

- ARM:** Mikroprozessor-Design, 1983 vom englischen Computerhersteller Acorn entwickelt
- Android:** freies, auf Linux basierendes Betriebssystem, insbesondere für Smartphones
- Assembler:** hardwarenahe, direkt in Maschinenbefehle übersetzbare Programmiersprache
- BIOS - Basic Input/Output System:** Computerfirmware, startet noch vor dem Betriebssystem
- Bash - Bourne-again shell:** eine freie Unix-Shell und Skript-Sprache
- Bluetooth:** Standard zur Funkübertragung von Daten
- C:** prozedurale Programmiersprache
- CC - Creative Commons:** NGO, die diverse Varianten von Standard-Lizenzverträgen zur Verfügung stellt, insbesondere freie Lizenzen
- CRF - Constant Rate Factor:** Standardqualitätseinstellung für die x264- und x265-Encoder. In dieser Arbeit wird der x264 Encoder via ffmpeg verwendet. Wertebereich zwischen 0 und 51, wobei niedrigere Werte zu einer besseren Qualität und höheren Datenraten führen.
- Codec:** Algorithmus zur Kodierung und Dekodierung von Signalen, im Kontext dieser Arbeit digitale Audio- und Videosignale, die komprimiert werden
- Compilieren:** Quellcode mittels Compiler in ausführbare Form (Binärcode) übersetzen
- Cron, Cronjob:** Cron ist eine Software die bei Unix-ähnlichen Betriebssystemen für die (regelmäßige) Ausführung von Prozessen verwendet wird. Ein Eintrag wird als Cronjob bezeichnet.
- DSI - Display Serial Interface:** Schnittstelle zur Anbindung von Displays
- Docker:** Programm zur Virtualisierung auf Betriebssystemebene (Containerisierung)

9. Zusammenfassung

Energy Harvesting: Energiegewinnung aus beispielsweise Wärmedifferenz, Vibration oder Luftströmungen für Geräte mit geringer Leistungsaufnahme

FD - File Descriptor: Konzept Unix-ähnlicher Betriebssysteme zum Öffnen, Erstellen und Bearbeiten von Dateien, die drei Standard-Datenströme stdin, stdout, stderr entsprechen den ersten 3 FD.

FIFO - First In - First Out: eine spezielle Art, Daten abzulegen und wieder abzurufen. Verhält sich ähnlich einem Rohr (vgl. Named Pipe), aus dem Material in der Reihenfolge in der es an einem Ende eingebracht wird am anderen ankommt.

FLAC - Free Lossless Audio Codec: verlustfreies Audiokompressionsformat und Container

FOSS - Free and Open Source Software: Unter einer Lizenz veröffentlichte Software, die ohne weitere Abgeltung jedermann bestimmte Rechte einräumt, wie beispielsweise Veränderung.

Fade-in, fade-out, cross-fade: Ein-, Aus- und Überblenden, beispielsweise eines Audiosignals beziehungsweise mehrerer Signale in- und übereinander.

FreeBSD: freies, Unix-ähnliches Betriebssystem

fstab - file system table: Konfigurationsdatei in Unix-ähnlichen Betriebssystemen, über die die zu verwendeten Dateisysteme konfiguriert werden können

Full HD: Auflösung von 1920x1080 Bildpunkten

GPIO - General Purpose Input/Output: Schnittstelle zur Ein- und Ausgabe von Daten, beispielsweise am RPI

GPL - GNU General Public License: freie Lizenz, insbesondere häufig für Software verwendet

H.264: auch als Advanced Video Coding (AVC) bekannter Standard zur Videokompression

H.265: auch als High Efficiency Video Coding (HEVC) bekannter Standard zur Videokompression

HDMI - High Definition Multimedia Interface: Schnittstelle für die digitale Bild- und Tonübertragung

HDR - High Dynamic Range: HDR Aufnahmen sind Aufnahmen, die große Helligkeitsunterschiede wiedergeben können als übliche Bilddateien mit 8 Bit je Farbkanal

9. Zusammenfassung

Hyperlapse: Technik der Zeitraffer-Fotografie für Bewegungsaufnahmen durch minimale Bewegung der Kamera zwischen jeder Aufnahme

I2C - Inter-Integrated Circuit: Digitale Schnittstelle

I2S - Inter-IC Sound: Eine Schnittstelle für Audiodaten

JPEG - Joint Photographic Experts Group: Gremium, das Komprimierungs- und Kodierungsmethoden vorschlägt, unter anderem das JPEG File Interchange Format, das aufgrund der üblichen Dateierdung .jpeg oder .jpg oft einfach als JPEG bezeichnet wird

Kernelmodul: im Betrieb ladbare Module, häufig für Gerätetreiber verwendet

LiPo - Lithium-Polymer-Akkumulator: Aktuelle Akku-Technologie

Linux: freies, Unix-ähnliches Betriebssystem

MOSFET - Metall-Oxid-Halbleiter-Feldeffekttransistor: ein Transistortyp der sich unter anderem als elektronischer Schalter eignet

MPPT - Maximum Power Point Tracking: Verfahren mit dem der Betriebspunkt bezüglich Leistungsabgabe einer Energiequelle – insbesondere einer Solarzelle – optimiert wird: Durch den nichtlinearen Zusammenhang von Spannung und Strom an der Solarzelle ergibt sich ein Punkt, an dem das Produkt, also die Leistung, maximal sind (Maximum Power Point). Dieser Punkt ist von Einflussfaktoren wie Sonneneinstrahlung und Temperatur abhängig. Es existieren verschiedene Techniken diesen Punkt zu verfolgen, etwa durch Variation der Last und Messung der Leistungsabgabe.

Making: iterativer, oft auch kollaborativer Prozess von „Basteln“ und Problemlösen mit Wurzeln in der DIY-Bewegung

Material eXchange Format - MXF: Dateiformat für Videodateien aus dem Rundfunk-Bereich

Matroska: freies Containerformat für Audio- und Videodateien

Named Pipe: Erweiterung des Pipeline-Konzepts, verhält sich als FIFO und stellt eine Methode zur Kommunikation zwischen Prozessen dar

OER - Open Educational Resources: freie Lern- und Lehrmaterialien unter offener Lizenz (etwa Creative Commons oder GNU General Public License)

OGG Vorbis: freies, verlustbehaftetes Audiokompressionsformat und Container

9. Zusammenfassung

POE - Power over Ethernet: Technik, um Energie über Ethernetdatenleitungen zu übertragen, beispielsweise durch Ausnutzen nicht verwendeter Leitungen oder Überlagerung von Signalleitungen mit einer Spannung.

PWM - Pulsweitenmodulation: eine (elektrische) Größe, im Kontext dieser Arbeit die elektrische Spannung, wird zwischen 0 und Versorgungsspannung mit einer bestimmten, auf die Verwendung abgestimmten ausreichend hohen Frequenz variiert, wobei der Tastgrad dieser Rechteckpulse die Leistung einstellt. Eine einfache Variante, bei der niedrige Frequenz aufgrund der hohen, in diesem Fall thermischen, Trägheit ausreicht, wäre das regelmäßige ein- und ausschalten einer Heizung, in Abständen die der gewünschten Temperatur entsprechen.

Pipe - Pipeline: Konzept Unix-ähnlicher Betriebssysteme um eine Folge von Prozessen durch ihre Ausgaben zu verketteten, so dass die Ausgabe jedes Prozesses (stdout) direkt als Eingabe (stdin) an den nächsten weitergeleitet wird: Mittels `—` kann die Ausgabe eines Programms einem anderen übergeben werden, `ls — wc -l` übergibt die Ausgabe von `ls` (Dateien in einem Verzeichnis auflisten) an `wc -l`, das die Zeilen zählt.

Pitch, tempo, speed: Tonhöhe, Tempo und Geschwindigkeit. Eine Änderung der Geschwindigkeit beim Abspielen eines Audiosignals ändert auch die Tonhöhe, bei einer Änderung des Tempos bleibt die Tonhöhe konstant. Ein höheres Tempo verkürzt die Dauer ohne die Tonhöhe zu erhöhen.

PNG - Portable Network Graphics: verlustfreies Grafikformat

Quicktime: Containerformat von Apple für Audio- und Videodaten

RGB: Der RGB-Farbraum wird aus einem Tripel von Intensitäten von Rot-, Grün- und Blauwerten aufgespannt

Ramdisk: Bereich im Arbeitsspeicher, der wie beispielsweise eine Festplatte angesprochen werden kann

Reflow-Ofen: Hauptkomponente beim Reflow-Löten, mit dem SMD-Bauteilen gelötet werden können

SSD - Solid-state Drive: Nichtflüchtiger Datenspeicher, üblicherweise mit deutlich höheren Schreib- und Lesegeschwindigkeiten als Festplatten

Signal 2, 9, 15: drei von vielen Signalen mit denen bei Unix-ähnlichen Betriebssystemen mit Prozessen kommuniziert werden kann, die dazu dienen, Prozesse zu beenden

9. Zusammenfassung

THD - Total Harmonic Distortion: Anteile der nichtlinearen Verzerrungen eines Signals

USV - Unterbrechungsfreie Stromversorgung: Technik um (kurzzeitige) Ausfälle der Energieversorgung zu überbrücken, indem elektrischer Strom beispielsweise aus einer Batterie, einem Akku, einem Kondensator oder über Energieumwandlung aus mechanischer (Schwungrad) oder chemischer (Diesel, Benzin, Wasserstoff) Energie zur Verfügung gestellt wird.

Upcycling: Umwandlung oder Wiederverwendung von Abfall, nutzlosen oder unerwünschten Produkten in neue Materialien oder Produkte. Anders als bei klassischer Müllverwertung kommt es bei dieser Form des Recyclings zu einer stofflichen Aufwertung.

WAV - Waveform Audio File Format: Containerformat zur digitalen Speicherung von Audiodaten

WLAN - Wireless Local Area Network: Lokales Funknetz, oft auch Wi-Fi

WSOLA - Waveform Similarity and Overlap Add: Methode zur Änderung des Tempos eines Audiosignals

Windows IoT Core: proprietäres, auf Windows basierendes Betriebssystem, optimiert für leistungsschwache Industrie-PCs, auch am RPI lauffähig

Literatur

- Albers, Erik u. a. (2015). »Positionspapier: Der Weg zur Stärkung freier Bildungsmaterialien«. In: URL: <http://buendnis-freie-bildung.de/positionspapier-%20er/> (siehe S. 19).
- Carvalho, R. u. a. (Juni 2017). »Real-time transmission of time-lapse imagery of glaciers in the southern Andes«. In: *2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE)*, S. 1–3. DOI: 10.1109/GRSS-CHILE.2017.7996019 (siehe S. 9).
- Driedger, Jonathan (2011). »Time-Scale Modification Algorithms for Music Audio Signals«. Magisterarb. URL: www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/01-students/2011_DriedgerJonathan_TSM_MasterThesis.pdf (siehe S. 17).
- DuBois, R. Luke (Aug. 2011). »Time-lapse phonography and the visual processing of music«. In: *Journal of Visual Culture* 10.2, S. 247–250. ISSN: 1470-4129. DOI: 10.1177/1470412911402910 (siehe S. 16).
- Fairbanks, G., W. Everitt und R. Jaeger (Jan. 1954). »Method for time of frequency compression-expansion of speech«. In: *Transactions of the IRE Professional Group on Audio AU-2.1*, S. 7–12. ISSN: 2168-2976. DOI: 10.1109/T-SP.1954.28213 (siehe S. 17).
- Fox, Matthew u. a. (2015). »Time-lapse thermography for building defect detection«. In: *Energy and Buildings* 92, S. 95–106. ISSN: 0378-7788. DOI: <https://doi.org/10.1016/j.enbuild.2015.01.021>. URL: <http://www.sciencedirect.com/science/article/pii/S0378778815000274> (siehe S. 9).
- Fricke, Ron (1992). *Baraka*. USA. URL: <http://www.documentarytube.com/videos/baraka> (siehe S. 12).
- Grandl, Maria (2017). »Offen lizenzierte Lehr- und Lernunterlagen für den (Einsatz im) Informatikunterricht«. Magisterarb. (siehe S. 18).
- Green, H. (Mai 1926). »Apparatus for Time Lapse Motion Picture Photography«. In: *Transactions of the Society of Motion Picture Engineers* 10.26, S. 147–151. ISSN: 0096-6460. DOI: 10.5594/J00800 (siehe S. 11).

Literatur

- Harris, M. P. (1982). »Promiscuity in the Shag as shown by timelapse photography«. In: *Bird Study* 29.2, S. 149–154. DOI: 10.1080/00063658209476750. eprint: <https://doi.org/10.1080/00063658209476750>. URL: <https://doi.org/10.1080/00063658209476750> (siehe S. 9).
- Higgins, M. (2017). *Time-lapse Photography: Art and Techniques*. Crowood Press, Limited. ISBN: 9781785002090. URL: <https://books.google.at/books?id=FmIBvgAACAAJ> (siehe S. 12).
- Johnston, Steven J und Simon J Cox (2017). »The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams«. In: *Electronics* 6.3. ISSN: 2079-9292. DOI: 10.3390/electronics6030051. URL: <http://www.mdpi.com/2079-9292/6/3/51> (siehe S. 5, 13, 28).
- Landecker, Hannah (Okt. 2009). »Seeing things: from microcinematography to live cell imaging«. In: *Nature Methods* 6. URL: <http://dx.doi.org/10.1038/nmeth1009-707> (siehe S. 9).
- Martin-Brualla, Ricardo, David Gallup und Steven Seitz (2015). »Time-lapse mining from internet photos«. In: *ACM Transactions on Graphics (TOG)* 34.4, S. 1–8 (siehe S. 12).
- Müller, Martin (2016). »Making im Unterricht am Beispiel des Raspberry Pi«. Magisterarb. (siehe S. 5).
- North, Nate (2008). *Silicon Valley Timelapse*. URL: <https://www.imdb.com/title/tt1254346/> (siehe S. 12).
- Plasser, Josef (2017). »Auswirkungen einer aufsteckbaren Always-on-Kamera auf die Gesellschaft«. Magisterarb. (siehe S. 18).
- Reinhard, Erik (2010). *High dynamic range imaging: acquisition, display, and image-based lighting*. 2.;2nd;2; Amsterdam [u.a.]: Elsevier (siehe S. 71).
- Schramm, U. (2001). »Lexikon der Neurowissenschaft in vier Bänden, Spektrum Akademischer Verlag (2000) ISBN 3-8274-0302-2«. In: URL: <https://www.spektrum.de/lexikon/neurowissenschaft/aufloesungsvermoegen/1071> (siehe S. 9).
- Tursun, Okan Tarhan u. a. (2015). »The State of the Art in HDR Deghosting: A Survey and Evaluation«. In: *Computer Graphics Forum*. DOI: 10.1111/cgf.12593 (siehe S. 72).
- Verhelst, W. und M. Roelands (Apr. 1993). »An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech«. In: *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Bd. 2, 554–557 vol.2. DOI: 10.1109/ICASSP.1993.319366 (siehe S. 17).
- Vlaj, Gernot (2014). »Machbarkeitsstudie zum Einsatz, zur Umsetzung von Schulbüchern als Open Educational Resource«. Magisterarb. (siehe S. 18).
- Weston, Chris (2015). *Spanning Time: The Essential Guide to Time-lapse Photography*. Abingdon: Focal Press (siehe S. 10–12).

Literatur

Zeller, Peter (2009). »Psychoakustik«. In: *Handbuch Fahrzeugakustik: Grundlagen, Auslegung, Berechnung, Versuch*. Hrsg. von Peter Zeller. Wiesbaden: Vieweg+Teubner, S. 151–157. ISBN: 978-3-8348-9322-2. DOI: 10.1007/978-3-8348-9322-2_7. URL: https://doi.org/10.1007/978-3-8348-9322-2_7 (siehe S. 16).