



SEARCH WING - TEILPROJEKT:
MESSUNG DES ENERGIEVERBRAUCHS EINES
RASPBERRY PI UND CHARAKTERISIERUNG DES
KAMERASYSTEMS

Faculty of Electrical Engineering
at Augsburg University of Applied Sciences

Systems Engineering 2

eingereicht von

Stephan Keller & Kilian Dierig & Leon Ranke

Betreuender Professor: Prof. Friedrich Beckmann

Abstract

Diese Arbeit ist Teil der Projektarbeiten des SearchWing-Projekts, in welchem es darum geht, mit Hilfe einer Flugdrohne (mit integrierter Kamera) im Mittelmeer in Seenot geratene Boote zu identifizieren. In diesem Teilprojekt werden zwei Teilaspekte untersucht:

Die Kamera wird von einem Raspberry Pi gesteuert. Das System benötigt Energie, welche die Reichweite der Drohne reduziert. Deswegen wird im ersten Teil dieser Untersuchung der Energieverbrauch des Raspberry Pi in unterschiedlichen Betriebsmodi (Leerlauf mit/ohne WLAN und Schießen von Fotos) analysiert. Es zeigt sich, dass die Unterschiede zwischen Leerlauf mit und ohne WLAN sehr gering ausfallen. Zufällige Schwankungen haben einen größeren Einfluss, als das Ausschalten des WLANs. In dieser Arbeit wird gezeigt, dass das Schießen von Fotos im png-Format sehr lange dauert und deshalb die JPEG-Komprimierung vorteilhaft ist. Das Schießen von mehreren Fotos pro Sekunde ist möglich, allerdings muss die Zeit zum Initialisieren der Kamera berücksichtigt werden.

Im zweiten Teil dieser Arbeit wird die Kameraauflösung untersucht. Dabei liegt das Augenmerk auf der Erkennung möglichst kleiner Objekte. Um das bestehende System, das V2 Kameramodul von Raspberry Pi, vergleichen zu können, wird der Test zusätzlich mit dem HQ Kameramodul und der Canon EOS600 Spiegelreflexkamera durchgeführt. Der Test unterteilt sich in zwei Teilbereiche. Zum einen ein Test mit kleinen Entfernungen von 2 bis 20 m. Dabei wird ein Testbild fotografiert und nachgeschaut, welches Muster noch zu erkennen ist. Zum anderen ein Test mit größeren Entfernungen von 10 bis 100 Metern, wobei dieser Versuch hauptsächlich zur Verifizierung des linearen Verlaufes der Kennlinie dient. Die gemessenen Werte werden mit den theoretisch errechneten Werten verglichen. Hierbei ergibt sich eine Differenz von ca. 60 % der gemessenen Werte. Dies liegt an mehreren Faktoren. Zum einen besitzt der Messaufbau Messfehler. Zum anderen kann es am sogenannten Bayern-Pattern des Sensors, sowie dessen Interpolationsverfahren liegen. Dies konnte jedoch aus zeitlichen Gründen und fehlendem Fachwissen nicht genauer analysiert werden.

Das Projektteam, bestehend aus Stephan Keller, Kilian Dierig und Leon Ranke, erklärt sich damit einverstanden, den Bericht im Rahmen des SearchWing-Projekts zu veröffentlichen.

Inhaltsverzeichnis

Abstract	I
Inhaltsverzeichnis	II
1 Untersuchung des Stromverbrauchs des Raspberry Pi	1
1.1 Messgeräte und Messaufbau	1
1.1.1 Raspberry Pi	1
1.1.2 Oszilloskop Tektronix	2
1.1.3 Verschaltung und sonstige Messgeräte	2
1.2 Ziel des Versuchs und Versuchsdurchführung	4
1.2.1 Messgrößen zur Berechnung des Ergebnisses	4
1.2.2 Untersuchung des Systems im Ruhezustand	5
1.2.3 Energieverbrauch der Kamera	5
1.3 Messergebnisse	6
1.3.1 Messung des Energiebedarfs im IDLE-Modus	6
1.3.2 Messung des Stromverbrauchs beim Foto schießen	8
1.4 Diskussion der Ergebnisse	12
1.4.1 Bewertung der Messungen des IDLE-Modus	12
1.4.2 Bewertung der Messung des Stromverbrauchs zum Schießen von Fotos	13
2 Untersuchung der Kameraauflösung	15
2.1 Messgeräte und Messaufbau	15
2.1.1 Raspberry Pi	15
2.1.2 V2 Kamera Modul	16
2.1.3 HQ Kamera Modul	16
2.1.4 Canon EOS 600D Kamera	17
2.1.5 Messaufbau	17
2.2 Theoretische Grundlagen	20
2.3 Versuchsdurchführung	23
2.3.1 Messung im Bereich bis 20 m	23
2.3.2 Messung im Bereich bis 100 m	24
2.4 Messergebnisse	25
2.4.1 Auswerten der Ergebnisse	25
2.4.2 Messung im Bereich bis 20 m	26
2.4.3 Messung im Bereich bis 100 m	31
2.5 Diskussion der Ergebnisse	34
3 Schlussfolgerung und Fazit	35

1 Untersuchung des Stromverbrauchs des Raspberry Pi

Die durchgeführte Untersuchung hat den Zweck, den genauen Stromverbrauch des Raspberry Pi in verschiedenen Betriebsszenarien zu untersuchen. Die Kenntnis über den Energieverbrauch des Systems ist von Interesse, weil die Energie für den Mikrocomputer aus der Boardbatterie entnommen wird und dadurch die Reichweite mit beeinflusst. Außerdem führt der Betrieb des Mikrocomputers zur Erwärmung des Systems. Da das System thermisch gut gegenüber der Umgebung geschützt ist, führen bereits geringe Wärmeemittenten dazu, dass sich das System deutlich erwärmt. Deshalb wird die Wärmeentstehung in verschiedenen Betriebsmodi untersucht. Besonders relevant sind hier insbesondere der Betrieb im Leerlauf (im weiteren IDLE-Verlauf genannt) und der Betrieb während des Schießens von Fotos. Dieses Kapitel beschreibt das Vorgehen zur Messung und analysiert auch die Ergebnisse der Untersuchung.

1.1 Messgeräte und Messaufbau

Für die Untersuchung des Stromverbrauchs wird ein Messaufbau entwickelt, mit welchem der Strombedarf des Systems gemessen werden kann. In diesem Kapitel werden die Einstellungen des Raspberry Pis und die verwendeten Messgeräte, sowie der Versuchsaufbau vorgestellt.

1.1.1 Raspberry Pi

Für den Versuch wird ein Raspberry Pi Zero W verwendet. Dieser verfügt über ein WLAN-Modul, sodass das Board mit einem WLAN-Netzwerk verbunden werden kann und dadurch über eine SSH-Verbindung ferngesteuert werden kann. Die wichtigsten Informationen aus dem Datenblatt sind in Tabelle 1.1 abgebildet.

Als Software wird das Kommandozeilenbasierte Raspberry Pi OS Lite installiert. Dieses System verfügt über keine GUI-Steuerung, weshalb es nur eine geringe Leistung im Betrieb benötigt. Für die Flugdrohne wird dieses OS auch verwendet, weshalb eine Vergleichbarkeit gut gegeben ist. Der Raspberry Pi wird über einen Micro-USB-Anschluss mit Energie versorgt. Das Betriebssystem wird auf der SD-Karte vorkonfiguriert und dann direkt von dieser gestartet. An den Pi ist noch eine Kamera angeschlossen: Die Raspberry Pi Camera Module V2 mit einem Kamerasensor von Sony (Sony IMX219 8-Megapixel). Der Anschluss der Kamera erfolgt über die 15-polige serielle MIPI-Schnittstelle (CSI-2). Zu Testzwecken wird an den Raspberry Pi teilweise noch über einen USB-Hub eine Tastatur verbunden und über den Mini-HDMI ein Bildschirm angeschlossen. Für die eigentlichen Messungen erfolgt die Kommunikation aber über die SSH-Schnittstelle des Pis. Dazu wird auf einem Desktop-Rechner die

Tabelle 1.1: Überblick über wichtigste Spezifikationen des Raspberry Pi Zero W [1]

Maße:	65mm x 30mm x 5mm
CPU:	Broadcom BCM2835 1Ghz
RAM:	512 MB
WLAN:	2,4 GHz 802.11n Wireless LAN
Bluetooth:	Bluetooth 4.0 und Bluetooth LE
Stromversorgung:	5V via Micro-USB-Anschluss
Video/Audio:	1080P HD Video & Stereo-Audio via mini-HDMI-Anschluss
Speicher:	Externe SD-Karte (SAN DISK extreme mit 64 GB)

Software Putty installiert, die durch aufrufen der IP-Adresse des Raspberry Pi nach Eingabe des Passworts diesen vollständig kontrollieren kann.

1.1.2 Oszilloskop Tektronix

Zur Messung des Stromverbrauchs wird ein Oszilloskop des Typs Tektronix TDS 1001-EDU mit zwei Kanälen verwendet. Das Gerät besitzt die Funktion, die Daten des aktuellen Messverlaufs zu speichern und als CSV-Datei zu exportieren, sodass diese mit Excel an einem Computer eingelesen werden kann. Das Oszilloskop ist deswegen nötig, da es im Vergleich zu einem Multimeter die zeitliche Änderung der Messwerte besser dokumentieren kann. Es wird insbesondere die Funktion genutzt, die letzten 10 Sekunden (bzw. 25 Sekunden) der Messung aufzeichnen zu können. Dadurch ist es dann möglich, den Anstieg des Stromverbrauchs während dem Schießen eines Fotos zu dokumentieren. Außerdem kann mit der Trigger-Funktion der Anstieg des Stroms richtig dokumentiert werden, sodass der richtige Bereich gespeichert wird.

1.1.3 Verschaltung und sonstige Messgeräte

Da der Energiebedarf des Systems gemessen werden soll, muss der Strombedarf und die Höhe der Versorgungsspannung jederzeit bekannt sein. Die Energieversorgung des Raspberry Pi erfolgt über ein Micro-USB-Kabel. Dieses wird aufgeschnitten, sodass der Plus- und der Minuspol des Kabels frei liegt. Der Strom kann nicht direkt gemessen werden. Stattdessen wird in Reihe zu dem Minus-Kabel ein $1,8 \Omega$ -Shunt-Widerstand eingebaut (gemessener tatsächlicher Widerstand über Vierleiter-Methode: $1,77 \Omega$). Fließt nun ein Strom durch den Shunt-Widerstand, fällt an diesem eine Spannung ab, die gemessen werden kann. Die Spannung wird mit dem Oszilloskop gemessen. Die Plusseite des USB-Kabels und die zweite Seite des Widerstands werden nun an ein Labornetzteil angeschlossen. Das Labornetzteil vom Typ LA-VOLTA DC- Power Supply BPS-305 stellt eine Konstantspannung bereit, die unabhängig von der Last ist. Eigentlich benötigt der Raspberry Pi laut Datenblatt eine Versorgungsspannung von $5 \text{ V} \pm 0,25 \text{ V}$ [2].

Da am Shunt-Widerstand aber auch eine Spannung abfällt, muss die Spannung des Labornetzteils höher ausfallen. Als praxistauglich hat sich hier eine Spannung von $5,5 \text{ V}$ erwiesen.

Dadurch liegt die Spannung die am Pi ankommt deutlich über 5 V (teilweise sogar über 5,25 V), niedrigere Spannungen haben aber regelmäßig zum Abstürzen des Pi geführt. Die Abstürze traten immer dann auf, wenn der Pi kurzzeitig mehr Strom benötigte. In diesem Moment steigt die Spannung am Shunt-Widerstand deutlich an, was dazu führt, dass die Spannung am Pi zusammenbricht und unter die untere Grenze von 4,75 V fällt. Erst mit einer Versorgungsspannung von 5,5 V konnte dieses Problem gemindert werden. Die Strompeaks treten zudem immer beim Booten auf. Deshalb wurde während des Boot-Vorgangs der Shunt-Widerstand kurzgeschlossen, sodass nur der Leitungswiderstand abfällt.

Zur Kontrolle der Spannung des Labornetzteils und der Spannung, die am Raspberry Pi direkt vor den Klemmen ankommt, wurden zwei Multimeter des Typs Agilent U1231A True RMS Multimeter eingesetzt. In Abbildung 1.1 ist ein Foto des Versuchsaufbaus zu erkennen. In Abbildung 1.2 ist der zugehörige Schaltplan des Versuchsaufbaus abgebildet.

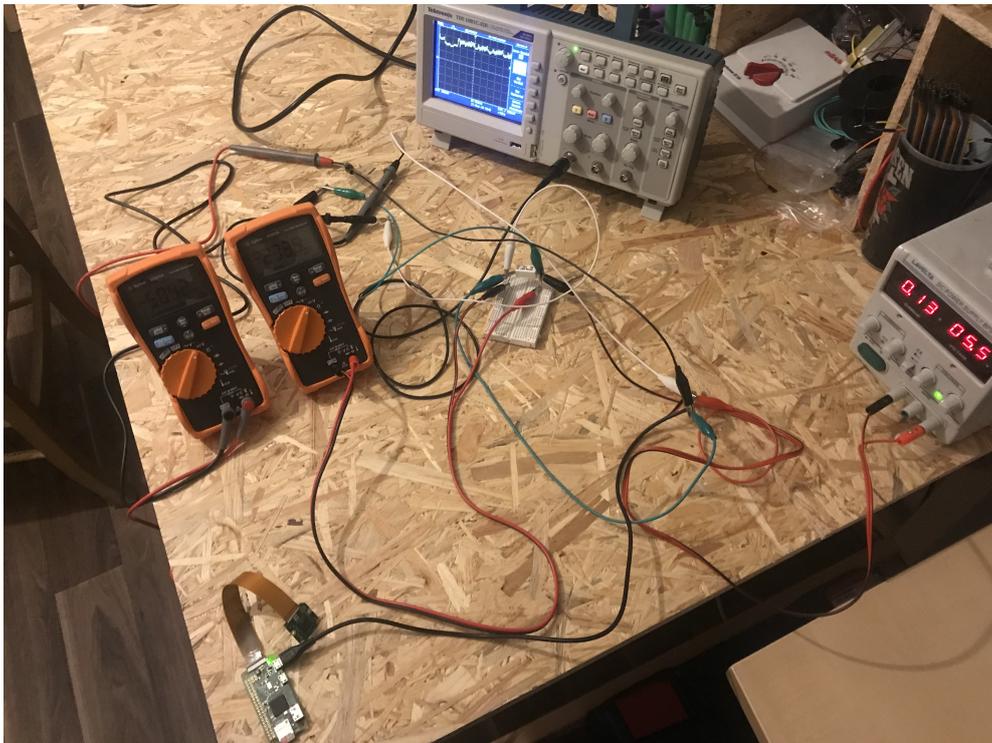


Abbildung 1.1: Versuchsaufbau mit Raspberry Pi und Shunt-Widerstand an Labornetzteil

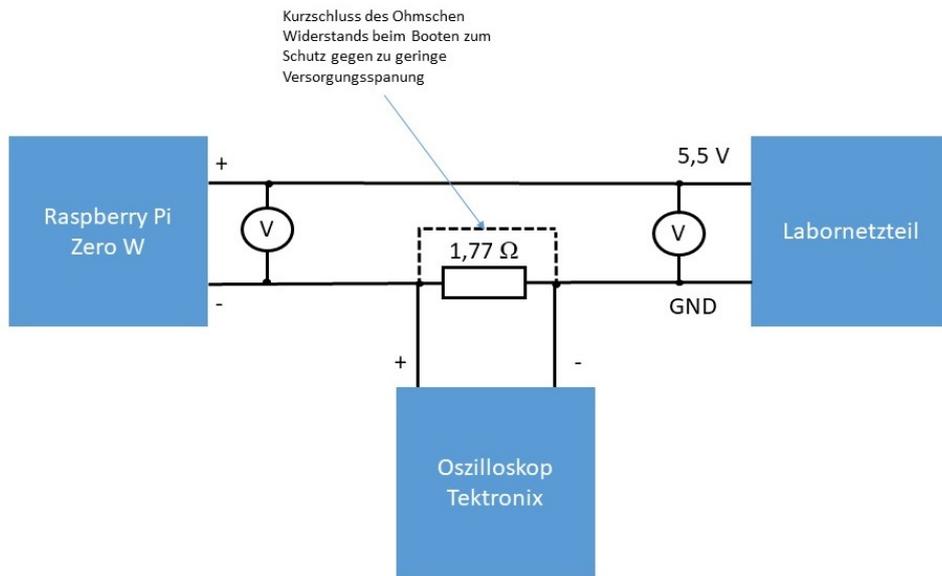


Abbildung 1.2: Verschaltung des Versuchsaufbaus als Schaltplan

1.2 Ziel des Versuchs und Versuchsdurchführung

In diesem Kapitel wird beschrieben, was genau untersucht wird und wie die Messungen durchgeführt werden, um die Ziele zu erreichen.

1.2.1 Messgrößen zur Berechnung des Ergebnisses

Für alle Messungen werden die gleichen Messgeräte zum Ermitteln der Ergebnisse verwendet, um statische Messfehler zu vermeiden. Es sollen im Wesentlichen folgende Größen bestimmt werden:

- **Gesamtstrom:** dieser wird berechnet, indem über das Ohmsche Gesetz aus dem Spannungsabfall am Shunt-Widerstand der Strom berechnet wird ($I = U_{Shunt}/R_{Shunt}$)
- **Zugeführte Leistung:** die zugeführte Leistung wird aus dem Produkt aus zugeführtem Gesamtstrom und der Versorgungsspannung bestimmt. Es ist zu beachten, dass Leitungsverluste und die Verluste am Shunt-Widerstand hier mit berechnet werden.
- **Genutzte Leistung:** hier wird die Leistung, die über dem Shunt-Widerstand abfällt, abgezogen. Es verbleiben lediglich die Leitungsverluste.

1.2.2 Untersuchung des Systems im Ruhezustand

Im ersten Test soll untersucht werden, wie groß der Energieverbrauch im Ruhezustand ist. Es werden drei Szenarien untersucht: der IDLE-Zustand ohne Wlan (Bluetooth auch ausgeschaltet), also auch ohne Verbindung über SSH, der IDLE-Zustand mit aktiviertem Wlan (Bluetooth an), der auf der Suche nach einer Verbindung ist. Und als drittes mit verbundenem Wlan (und aktivem Bluetooth). Das Wlan kann mit dem Befehl „rfkill block 0“ abgeschaltet werden. Es ist zu beachten, dass es sich hierbei lediglich um eine softwareseitige Abschaltung handelt (gleiches gilt für Bluetooth). Ein hardwareseitiges Abschalten ist nicht möglich.

Bei der Messung ohne Wlan wird das System zunächst über SSH mit Wlan gestartet und softwareseitig das Wlan ausgeschaltet. Dies führt dazu, dass ein Fernsteuern nicht mehr möglich ist. Nach Abschalten wird mit dem Oszilloskop der durchschnittliche Energieverbrauch berechnet. Bei der Messung mit aktiviertem, aber unverbundenem Wlan werden über SSH die Zugangsdaten zum Wlan auf einen Namen geändert, der nicht existiert, sodass das System das Netz nicht findet. Nach einem Neustart versucht das System sich mit diesem nicht vorhandenen System zu verbinden. Dieser Vorgang wird wieder mit dem Oszilloskop dokumentiert.

Bei der Messung mit aktiviertem und verbundenem Wlan wird einfach der Energieverbrauch im verbundenen Zustand mit dem Oszilloskop aufgezeichnet.

1.2.3 Energieverbrauch der Kamera

Die Messung des Energieverbrauchs der Kamera erfolgt mit Hilfe eines Programms, welches über die Kommandozeile des Raspberry Pi aufgerufen wird. Das Programm ist für alle Versuche ähnlich aufgebaut: Zunächst wird das Kameramodul initialisiert. Anschließend wird in einer for-Schleife eine feste Anzahl Bilder geschossen. Die Aufnahmen werden dabei nicht auf der SD-Karte gespeichert, sondern stattdessen in einem speziellen RAM-Speicher abgelegt. Dies liegt daran, dass das Abspeichern auf der SD-Karte viel Zeit benötigt und es für die Untersuchungen des Bildes nicht nötig ist, dass das Bild später wieder abgerufen werden kann. Das bedeutet, dass die Bilddaten beim Ausschalten des Systems nicht gespeichert werden und verloren gehen.

Die Bilder werden in der höchstmöglichen Auflösung geschossen: 3280x2464 Pixel. Für den späteren Einsatz der Drohne ist die JPEG-Komprimierung der Standard. Deshalb werden grundsätzlich die Bilder als JPEG-Datei komprimiert. Allerdings gibt es auch eine Messung, in welchem zum Vergleich die Bilder nicht komprimiert gelassen werden und als png-Datei gespeichert werden.

Zur besseren Einteilung, welcher Zeitbereich welchem Prozess (z.B. Anschalten des Kameramoduls) zuzuordnen ist, wird das Programm angepasst, sodass einmal nur ein Bild aufgenommen wird und einmal sodass fünf bzw. drei Bilder in Reihe aufgenommen werden. Zusätzlich werden für die Messungen im Programm teilweise Pausen eingefügt, mit welchen die einzelnen Teilprozesse der Software besser zu trennen ist. In Abbildung 1.3 ist exemplarisch der Code zum Schießen von drei Bildern dargestellt, wobei nach jedem Fotoschuss eine Pause von einer Sekunde eingehalten wird.

```
GNU nano 3.2          foto.py          Modified
from io import BytesIO
from picamera import PiCamera
from time import sleep

my_stream=BytesIO()
camera = PiCamera()
camera.resolution=(3280, 2464)
camera.start_preview()

#sleep(3)

for i in range(3):
    camera.capture('/mnt/ramdisk/image%s.jpeg' %i, format='jpeg', use_video_port=False)
    sleep(1)
    print("Ein Foto gemacht und gespeichert")
camera.stop_preview()
```

Abbildung 1.3: Exemplarischer Programmcode zum Schießen von drei Bildern in Reihe mit anschließender Pause von einer Sekunde

1.3 Messergebnisse

Dieses Kapitel stellt die Messergebnisse zum Energieverbrauch vor.

1.3.1 Messung des Energiebedarfs im IDLE-Modus

Zunächst werden die Ergebnisse der Messungen des IDLE-Modus vorgestellt. In Tabelle 1.2 sind die gemittelten Werte der Strommessung des IDLE-Modus zu finden.

IDLE-Modus ohne Wlan

Im IDLE-Modus ohne Wlan (und ohne Bluetooth) ist die SSH-Verbindung mit dem Desktop-Computer getrennt. Es findet keine Kommunikation mit dem Raspberry Pi statt. Alle Peripheriegeräte sind abgehängt. Es ist lediglich die Kamera angeschlossen (jedoch nicht angeschalten). Es werden zwei Messungen dazu durchgeführt. Die erste Messung listet den Strombedarf direkt nach dem Ausschalten des Wlans und Bluetooth auf. Hier wird ein mittlerer Stromverbrauch von ca. 84 mA gemessen. Dies entspricht einer zugeführten Leistung 0,46 W bzw. einer genutzten Leistung von 0,45 W.

Bei der zweiten Messung wird der Pi neu gestartet. Eine Verbindung über SSH wird hier überhaupt nicht mehr hergestellt. Hier wird ein mittlerer (um Verluste bereinigter) Verbrauch von 0,50 W gemessen, liegt also tatsächlich etwas höher als zuvor.

IDLE-Modus mit verbundenem Wlan

Im zweiten Versuch ist der Raspberry Pi dauerhaft mit dem Internet verbunden und es besteht auch eine aktive SSH-Verbindung. Der Stromverbrauch liegt allerdings nicht höher, als mit

ausgeschaltetem Wlan. Es wird ein mittlerer Stromverbrauch von 84 mA und entsprechend eine um den Verlust des Shunts bereinigte Leistung von 0,45 mW gemessen.

IDLE-Modus mit Wlan-Suche

Im dritten Versuch ist der Wlan-Chip aktiv, er ist allerdings nicht in der Lage eine Verbindung mit dem Netz herzustellen, da dem System ein falscher Netzname übergeben wurde. Es wurden wieder zwei Messungen durchgeführt. Die erste Messung liefert einen mittleren Strom von 87 mA, die zweite einen Strom von 85 mA. Entsprechend wird eine bereinigte Leistung von 0,47 bzw. 0,46 W benötigt. Während der Strom in den vorangegangenen Messungen über die Zeit überwiegend konstant war, ist bei dieser Messung ein regelmäßiger Ausschlag zu erkennen. Dieser ist in Abbildung 1.4 gut zu erkennen. Es liegt die Vermutung nahe, dass diese Peaks entstehen, weil das System alle fünf Sekunden versucht, sich ins Internet einzuwählen. Da diese aber nur sehr kurz sind, tragen die Peaks nur relativ wenig zum Gesamtenergieverbrauch bei.

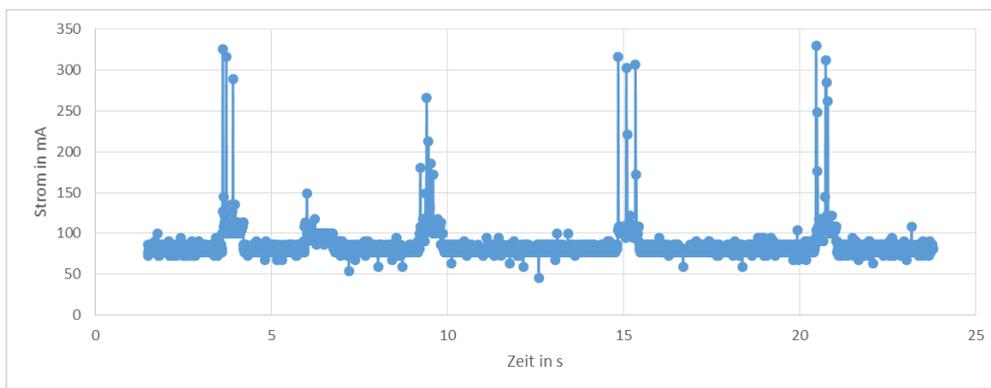


Abbildung 1.4: Stromverbrauch bei eingeschaltetem Wlan ohne Netzverbindung

Tabelle 1.2: Mittlerer Strom und mittlere Leistung der Messung des IDLE-Modus

	Mittlerer Strom [mA]	Mittlere bereinigte Leistung [W]
IDLE ohne WLAN (1. Messung)	83,8	0,45
IDLE ohne WLAN (2. Messung)	94,3	0,50
IDLE mit verbundenem WLAN	83,6	0,45
IDLE WLAN suchend (1. Messung)	87,4	0,47
IDLE WLAN suchend (2. Messung)	85,4	0,46

1.3.2 Messung des Stromverbrauchs beim Foto schießen

Die Messergebnisse zum Energiebedarf während des Schießens von Fotos werden in folgendem Abschnitt vorgestellt. In Tabelle 1.3 sind zusammenfassend die Messungen des durchschnittlichen Strom und Leistungsbedarfs der Fotomessungen. Es ist die mittlere Leistung über die Gesamtzeit, die das Programm benötigt, um die Fotos zu schießen, gegeben.

Tabelle 1.3: Mittlerer Strom und mittlere Leistung beim Schießen von Fotos

	Mittlerer Strom [mA]	Mittlere bereinigte Leistung [W]
Schießen eines Fotos	220	1,12
Schießen von 5 Fotos	229	1,17

Messung des Stromverbrauchs beim Schießen eines Fotos

Bei dieser Messung wird der Energiebedarf für das Schießen eines einzelnen Fotos bei einer Abspeicherung als JPEG-Format untersucht. Im Programm werden keine künstlichen Pausen gemacht.

Es wird ein mittlerer Strom von 220 mA gemessen. Dies entspricht einem Shunt-Bereinigten Stromverbrauch von 1,12 W. In Abbildung 1.5 ist der zeitliche Verlauf des Stroms beim Schießen eines Fotos gegeben.

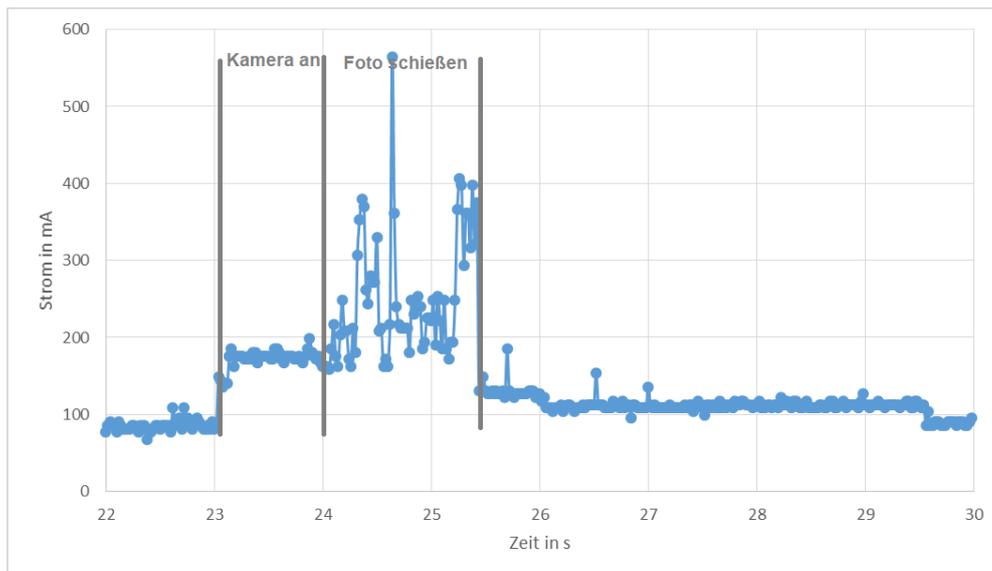


Abbildung 1.5: Anstieg des Stromverbrauchs während des Schießens eines einzelnen Fotos

Messung des Stromverbrauchs beim Schießen von mehreren Fotos

Analog zur Messung des Stromverbrauchs mit einem Foto wird die Messung wiederholt, wobei in diesem Fall fünf Bilder auf einmal geschossen werden. Alle anderen Parameter bleiben unverändert. Es wird weiterhin keine Pause gemacht und die Bilder werden wieder im JPEG-Format abgespeichert. In Abbildung 1.6 ist das Messergebnis des Stromes während der Messung von fünf Bildern über die Zeit dargestellt. Es wird ein mittlerer Strom von 229 mA benötigt. Der Pi benötigt während des Schießens der Bilder eine mittlere Leistung von 1,17 W.

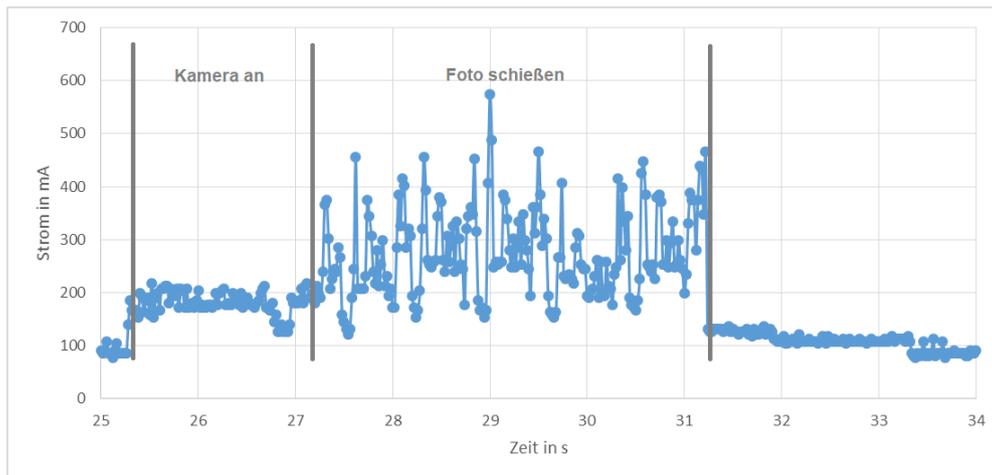


Abbildung 1.6: Anstieg des Stromverbrauchs während des Schießens von fünf Fotos direkt hintereinander

Messung des Stromverbrauchs beim nicht komprimierten Speichern als png

Bisher wurden die Daten komprimiert als JPEG-Datei abgespeichert. In einer Vergleichsmessung wird das Bild nicht komprimiert und stattdessen als png-Datei abgespeichert. Der Verlauf beim Schießen eines Fotos ist in Abbildung 1.7 zu erkennen. Die Messung ist einer der ersten gewesen, die durchgeführt wurden und konnte aufgrund der begrenzten Verfügbarkeit des Oszilloskops nicht wiederholt werden. Die Rahmenbedingungen unter denen dieser Ausschnitt aufgenommen wurde unterscheiden sich von den anderen Bildern: zunächst gibt es keine Verbindung mit dem Netz, was zu den bekannten Wlan-Peaks alle 5 Sekunden führt. Diese Peaks dominieren das Erscheinungsbild des Verlaufs. Des Weiteren ist für den Versuch permanent ein Bildschirm und eine Tastatur über einen USB-Hub angeschlossen, die Energie verbrauchen und den mittleren Stromverbrauch stark erhöhen. Allerdings kann dieser Mehrverbrauch durch eine Kompensationsmessung zumindest für die Mittelwerte herausgerechnet werden. Der um den Bildschirm und Tastaturverbrauch bereinigte mittlere Strombedarf liegt bei 347 mA und die bereinigte Leistung ohne Tastatur, Bildschirm und Shunt-Widerstand liegt bei 1,14 W. Da die Schwankungen der Messwerte im IDLE-Modus sehr groß waren und kein genauer

Unterschied zwischen Wlan-Suchend und deaktiviertem Wlan angegeben werden kann, wird der Wlan-Mehrverbrauch in der Berechnung ignoriert.

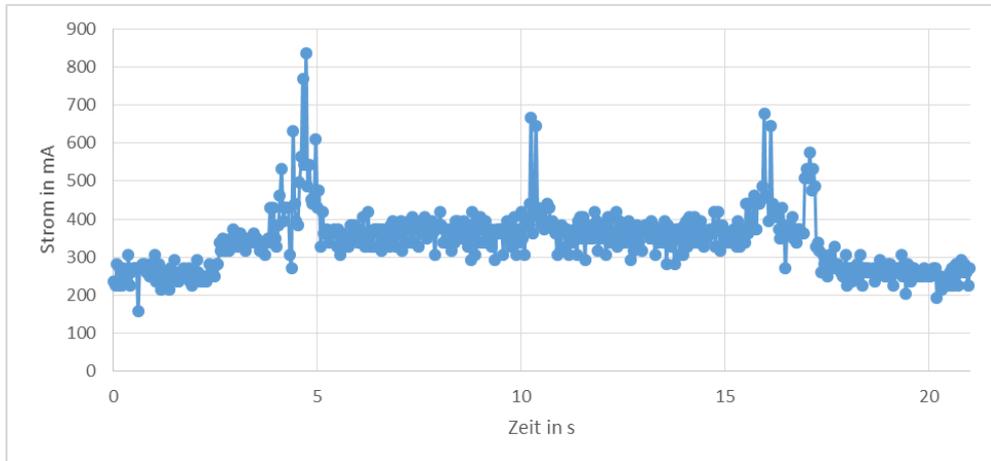


Abbildung 1.7: Stromverlauf bei Abspeicherung eines einzelnen Bildes als png-Datei

Stromverbrauch bei zusätzlichen Pausen

Zur besseren Abtrennung der einzelnen Phasen, die das Programm durchläuft, werden noch weitere Messungen durchgeführt, in welchen während des Fotoschießens Pausen eingelegt werden, um besser unterscheiden zu können, welcher Zeitanteil welcher Stromphase zuzuordnen ist. Bei diesen Messungen ist der durchschnittliche Strombedarf irrelevant, da er durch die extra Pausen verzerrt wird.

Stromverbrauch bei Messung von fünf Fotos mit zusätzlicher initialer Pause

Diese Messung entspricht exakt der Messung des Strombedarfs bei fünf Fotos hintereinander. Im Unterschied dazu wird allerdings nach dem Initialisieren und Einschalten der Kamera eine Pause von drei Sekunden eingelegt. In Abbildung 1.8 ist der zeitliche Verlauf dargestellt. Die Messdaten wurden mit der Trigger-Funktion des Oszilloskops aufgenommen, deswegen ist die Zeitachse hier mittig auf null.

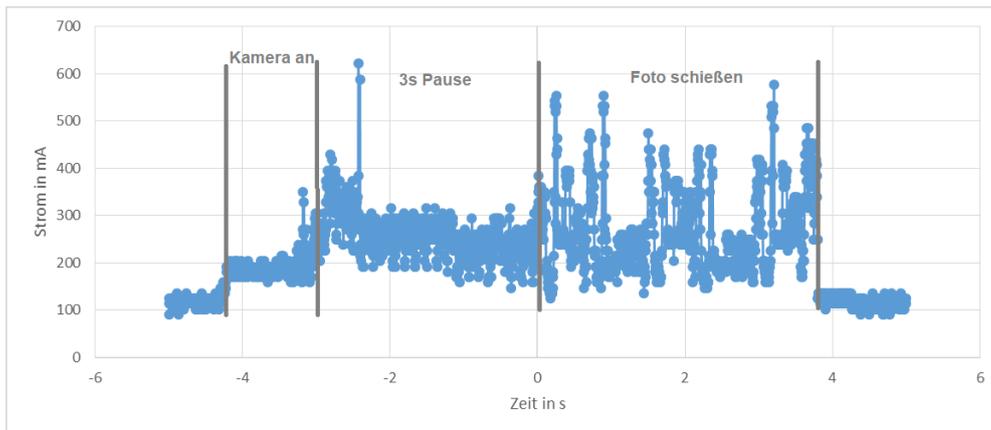


Abbildung 1.8: Messung von fünf Fotos hintereinander mit zusätzlicher Pause nach dem Einschalten der Kamera von 3 s

Stromverbrauch bei Messung von einem Foto mit zusätzlicher initialen Pause In Abbildung 9 ist die gleiche Messung noch einmal, wobei in dieser Messung anstelle von fünf Bildern nur ein Bild gemacht worden ist.

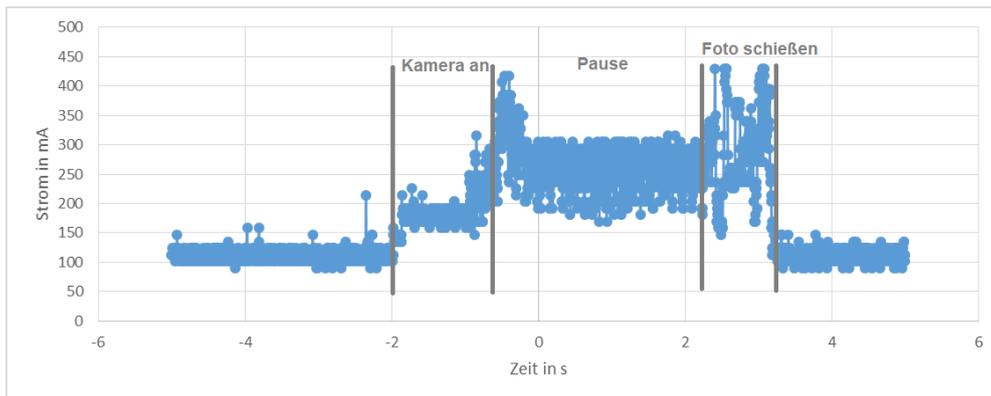


Abbildung 1.9: Messung eines Fotos mit zusätzlicher Pause nach dem Einschalten der Kamera von 3 s

Schießen von drei Fotos hintereinander mit 1s Pause zwischen den Bildern Zuletzt wird noch eine Messung durchgeführt, in welcher drei Fotos in Reihe geschossen werden, wobei nach jedem Foto eine Pause von einer Sekunde eingehalten wird. Der zeitliche Verlauf dieser Messung ist in Abbildung 1.10 gegeben.

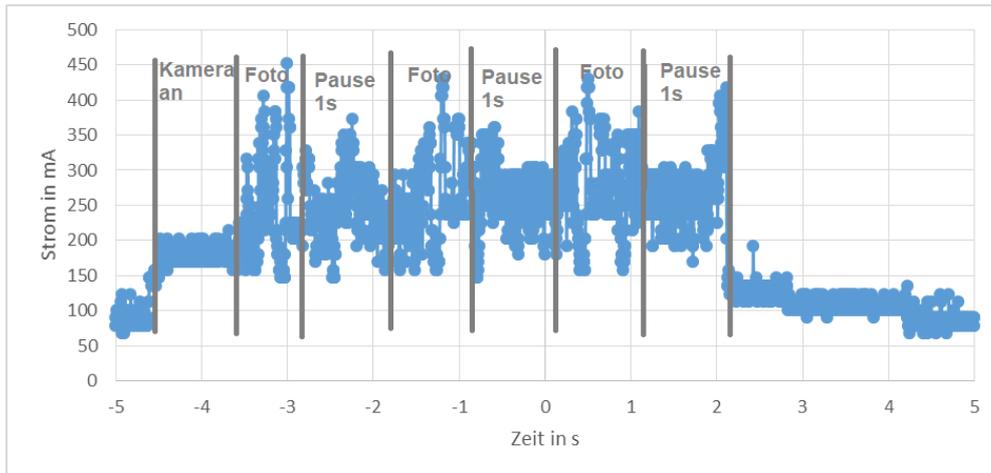


Abbildung 1.10: Stromverlauf beim Schießen von drei Bildern in Reihe mit einer Pause von einer Sekunde nach jeder Aufnahme

1.4 Diskussion der Ergebnisse

In diesem Abschnitt werden die Ergebnisse analysiert und miteinander verglichen.

1.4.1 Bewertung der Messungen des IDLE-Modus

Die Messergebnisse zum IDLE-Modus liegen alle in einer ähnliche Größenordnung. In der Messung konnten keine großen Unterschiede beim An- und Ausschalten des Wlans festgestellt werden. Die Daten deuten zudem darauf hin, dass der Stromverbrauch zufällig relativ stark schwankt, da bei der Messung ohne Wlan in einer zweiten Messung 12,5 % höher ausfällt (94,3 mA vs. 83,8 mA). Dass die Messungen Schwankungen unterworfen sind, ist nicht überraschend, da auf einem Mikrocomputer diverse Prozesse laufen, auf die man während einer Messung keinen Einfluss hat, die immer wieder zu einem kurzzeitigen Anstieg des Stroms führen können.

Messdaten von Nutzern aus dem Internet zeigen aber, dass sich die gemessenen Werte in der richtigen Größenordnung befinden. Auf der Seite bitblokes.de werden ebenfalls Messwerte vorgestellt. Hier liegt der IDLE-Strom bei ca. 100 mA ohne Wlan und 120 mA mit Wlan. Zu beachten ist hier allerdings, dass als Betriebssystem vermutlich nicht Raspi OS light, sondern das normale OS verwendet wurde (Raspi OS ist Standard, weshalb davon ausgegangen werden muss, die Webseite gibt selber keine Auskunft). Durch das konventionelle Betriebssystem ist entsprechend der höhere Stromverbrauch erklärbar [3].

Insgesamt werden bei der Messung des IDLE-Modus konstante Werte ermittelt. Für die weitere Auslegung der Drohne ist man bei der Auslegung des Energieverbrauchs mit einem Wert von 0,6 W auf der sicheren Seite (0,5 W maximaler gemessener Wert +0,1W Sicherheitsbeitrag).

1.4.2 Bewertung der Messung des Stromverbrauchs zum Schießen von Fotos

Die Ergebnisse des Stromverbrauchs beim Schießen von Fotos werden in diesem Kapitel diskutiert.

Zusätzlicher Energieverbrauch beim Schießen von Fotos

Der Stromverbrauch der beim Schießen von Bildern im JPEG-Format ermittelt wird, liegt in einer gleichen Größenordnung, wie er auch auf Technikwebseiten im Internet ermittelt wird [3].

Aus den zeitlichen Verläufen des Stromverlaufs lässt sich nun noch berechnen, welchen zusätzlichen Energieaufwand das Schießen eines JPEG-Fotos bzw. mehrerer JPEG-Fotos bedeutet. Dazu wird zunächst die Dauer aus den Diagrammen zum Strom-/Zeitverlauf aufgelistet. Bei der Messung mit einem Bild dauert der Fotografier-Prozesses sehr genau 3 s. Bei fünf Bildern hintereinander dauert das Schießen der fünf Bilder insgesamt 5,9 s. Auffallend ist hier, dass für die Bilder insgesamt eine sehr hohe Initialisierungszeit nötig ist. Das Schießen der einzelnen Bilder dauert dagegen dann nicht mehr lange.

In Tabelle 1.4 sind die Ergebnisse der Berechnung der Energiekosten nochmal dargestellt. Es ist nur der zusätzliche Energiebedarf berechnet, der Verbrauch des Leerlaufs ist hier bereits heraus gerechnet.

Der Verlauf des Schießens eines Fotos als png ist aufgrund der abweichend Testeinstellungen schwierig einzuordnen. Es kann dennoch festgehalten werden, dass das Schießen eines Fotos wesentlich länger dauert, als mit einer Abspeicherung als JPEG. Dadurch fällt der Energieverbrauch pro Foto deutlich höher aus. Für die Drohnenaufnahmen mit begrenzter Energieverfügbarkeit ist das Komprimieren als JPEG die bessere Wahl, solange die komprimierte Bildqualität ausreicht.

Tabelle 1.4: Energiekosten für das Schießen eines Fotos bzw. mehrerer Fotos

	Dauer Bild machen [s]	Energiebedarf [Ws]	Energie/Bild [Ws]
1 Bild (JPEG)	3,0	1,86	1,86
5 Bilder (JPEG)	5,9	3,95	0,79
1 Bild (PNG)	14,5	9,28	9,28

Analyse des qualitativen Stromverlaufs

Zuletzt soll der qualitative Verlauf des Stromverlaufs beim Schießen eines Fotos analysiert werden. Dazu fließen die Ergebnisse der Messungen mit Zusatzpausen ein. Der grundsätzliche Stromverlauf zu Schießen eines Fotos ist eigentlich immer ähnlich. Mit dem Beginn des Programms wird zunächst die Kamera initialisiert und eingeschaltet. Es ist davon auszugehen, dass der Sprung zu Beginn des Programms darauf zurückzuführen ist, dass die Kamera eingeschaltet ist. Anschließend bewegt sich der Strom in einer Seitwärtsbewegung für mehr als eine Sekunde ohne Änderungen fort. Dies gehört vermutlich weiterhin zum Einstellvorgang

der Kamera. In Abbildung 1.7 und 1.8 ist jeweils nach dem Initialisierung eine Pause von 3 s eingebaut. Diese ist hier aber kein seitwärtsverlauf, sondern die Pause wird mit einem Strompeak eingeleitet der abflacht. Die Ursache für den Peak ist nicht bekannt. Dennoch deuten die Messdaten darauf hin, dass die Einteilung in Initialisierung und Foto-Schieß-Abschnitt richtig ist.

Nach der Initialisierung geht die Messung in eine dynamische Phase über. Der mittlere Strom steigt nochmal und der Strom beginnt mit einigen Peaks deutlich zu springen. Hier werden jetzt kurzzeitig Spitzenströme von über 600 mA erreicht. In diesem Abschnitt werden vermutlich die einzelnen Fotos gemacht. Dies zeigt auch Abbildung 1.9, in welcher nach jedem Foto eine Pause von 1 s gemacht wird. Hier sind die ruhigen Phasen der Pause und die dynamischen Phasen abwechselnd, was bestätigt, dass die dynamischen Phasen durch das tatsächliche Schießen der Fotos ausgelöst wird.

Nach dem Ende des Programms bleibt der Stromverbrauch zunächst für einige Sekunden höher über dem Durchschnitt (gut zu sehen in Abbildung 1.6). Auch hier ist die Ursache für den überdurchschnittlichen Strombedarf unklar. Vermutlich ist er mit internen Prozessen des Pi zu erklären.

2 Untersuchung der Kameraauflösung

Im Rahmen des Searchwing-Projekts werden während eines Drohnen flugs Bilder aufgenommen. Dabei sollen Schiffe welche in Seenot geraten sind identifiziert werden. Die Drohne befindet sich hierbei auf eine Flughöhe von ca. 550 Metern über den Meeresspiegel. In solch hohen Distanzen ist es durchaus von Bedeutung welche Objekte von dem Kamerasystem erkannt werden können. Grundsätzlich stellt sich die Frage ob ein Schiffbrüchiges Boot noch erkannt werden kann.

Daher wird in diesem Teil der Arbeit das Kamerasystem genauer Charakterisiert. Dabei ist von Bedeutung welche physikalische Fläche ein Pixel auf dem geschossenen Foto entspricht. Hierbei handelt es sich auch automatisch um die kleinste erkennbare Größe für die Kamera. Aktuell ist in der Drohne das Raspberry Pi V2 Kamera Modul verbaut, ein umstieg zu dem Höherwertigen HQ (High Quality) Kamera Modul, welches ebenso mit dem Raspberry Pi kompatibel ist, wäre denkbar. Deshalb wird in diesem Abschnitt zusätzlich das HQ Kamera Modul untersucht um, im falle des Tausches, die Auswirkungen auf die Auflösung aufzuzeigen. Die Ergebnisse dieser beiden Module werden zusätzlich mit denen einer Spiegelreflexkamera verglichen. Dies soll die Ergebnisse der Raspberry Pi Module in Relation zu denen einer hochwertigen eigenständiger Kamera setzen.

2.1 Messgeräte und Messaufbau

Für die Untersuchung der Kameraauflösung wird ein Messaufbau entworfen, mit welchem die Kameraauflösung verifiziert werden kann. In diesem Abschnitt werden die Verwendeten Messgeräte und der Messaufbau beschrieben.

2.1.1 Raspberry Pi

Auch bei diesem Versuch wird ein Raspberry Pi Zero W wie in Abschnitt 1.1.1 beschrieben verwendet.

Als Software wird hierbei jedoch das „Raspberry Pi OS with desktop and recommended software“ verwendet. Diese wird verwendet, da die Software die Bedienung des Raspberry Pi deutlich erleichtert. Mit Hilfe des Pis wird die Kamera angesteuert und die Bilder gespeichert. Die Steuerung erfolgt hierbei erneut über eine SSH Verbindung.

2.1.2 V2 Kamera Modul

Das momentan in der Flugdrohne verbaute Kameramodul ist das Raspberry Pi V2 Modul. Dieses wird über die 15-polige serielle MIPI-Schnittstelle (CSI-2) an den Raspberry Pi angeschlossen. Das Modul benötigt kein Objektiv, wodurch es einfach zu bedienen ist. Die Belichtungszeit wird automatisch ermittelt und eingestellt. Die wichtigsten Daten sind in Tabelle 2.1 dargestellt.

Tabelle 2.1: Übersicht über wichtigste Spezifikationen des V2 Kamera Moduls[4]

Sensor	Sony IMX219
Auflösung [MP]	8 Megapixel
Sensor Auflösung [pixel]	3280 x 2464 pixel
Objektiv	keins
Horizontaler Öffnungswinkel	62, 2°
Vertikaler Öffnungswinkel	48, 8°

2.1.3 HQ Kamera Modul

Das High Quality (HQ) Kameramodul ist eine verbesserte Variante des V2 Moduls. Dieses ist ebenfalls speziell für den Raspberry Pi entwickelt und angepasst. Es wird ebenso über die 15-polige serielle MIPI-Schnittstelle (CSI-2) an den Raspberry Pi angeschlossen. Dieses Modul kann ohne Objektiv nicht verwendet werden, wodurch ein zusätzliches Bauteil benötigt wird. In diesem Versuch wird ein 6 mm Weitwinkelobjektiv von Raspberry Pi verwendet. Das scharf stellen mit Hilfe des Objektivs ist nur bis zu einer Entfernung von ca. 50 m nötig. Danach ist es auf das Maximum eingestellt und muss für größere Entfernungen nicht mehr verstellt werden. Wie beim V2 Kamera Modul wird die Belichtungszeit auch hier automatisch ermittelt und eingestellt. Die wichtigsten Daten des Moduls, sowie des Objektivs, sind in Tabelle 2.2 zusammengefasst.

Tabelle 2.2: Übersicht über wichtigste Spezifikationen des HQ Kamera Moduls [4, 5]

Sensor	Sony IMX477
Auflösung [MP]	13,3 Megapixel
Sensor Auflösung [pixel]	4056 x 3040 pixel
Objektiv	Offizielles Raspberry Pi HQ Kamera 6mm Weitwinkel-Objektiv
Horizontaler Öffnungswinkel	63°
Vertikaler Öffnungswinkel	49, 6°

2.1.4 Canon EOS 600D Kamera

Um die Raspberry Pi Kameramodule mit herkömmlichen Kamerasystemen vergleichen zu können, wird eine Canon EOS 600D verwendet. Diese Spiegelreflexkamera besitzt auch ein Weitwinkelobjektiv, wodurch die Öffnungswinkel der verschiedenen Kameras vergleichbar sind. Da die Canon Kamera viele Einstellmöglichkeiten bietet, auf die geachtet werden muss, wird die Kamera im Automatikmodus verwendet. Dadurch stellen sie die wichtigen Parameter selbst ein, wie z.B. die Belichtungszeit oder der Fokus.

Tabelle 2.3: Übersicht über wichtigste Spezifikationen der Canon Kamera [6, 7]

Sensor	CMOS-Sensor APS-C
Auflösung [MP]	18 Megapixel
Sensor Auflösung [pixel]	5184 x 3456 pixel
Objektiv	EF-S 18-55mm f/3.5-5.6 IS II
Horizontaler Öffnungswinkel	64,5°
Vertikaler Öffnungswinkel	45,5°

2.1.5 Messaufbau

Der Messaufbau besteht aus zwei Teilen. Zum einen aus dem Messstand, auf welchem die Kameras befestigt sind, und zum anderen aus einer Wand, welche das Testbild beinhaltet. Der Messstand ist dabei parallel zur Wand verschiebbar, wodurch der Abstand d verändert werden kann. Ein schematischer Aufbau ist Abbildung 2.1 zu entnehmen.

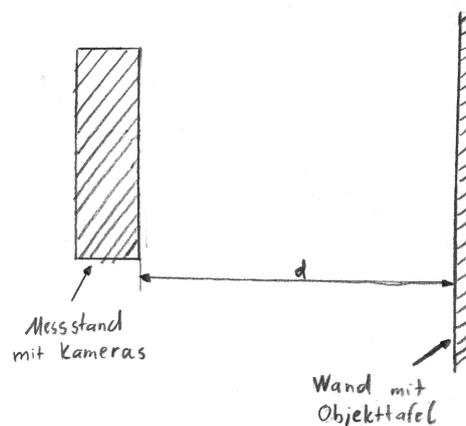


Abbildung 2.1: schematischer Messaufbau

Am Messstand sind alle drei Kameras (1) angebracht, wodurch mit allen gleichzeitig ein Bild erstellt werden kann. Zudem sind zwei Laptops (2) aufgebaut um mit Hilfe einer SSH Verbindung die Raspberrypi Zero W anzusteuern. Dazu wird von einem Laptop ein Hotspot erzeugt, in welchem sich alle anderen Geräte einwählen.

Um die Distanz zur Wand zu Ermitteln wird ein Distanzmessgerät (3) verwendet. Abbildung 2.2 zeigt den Aufbau des Messstandes. Die dritte Kamera ist hier nicht gezeigt, da das Bild mit ebendieser gemacht wurde.



Abbildung 2.2: Messstand

Für das schießen eines Fotos der Kameramodule mit Hilfe eines Raspberry Pi wird ein einfaches Python Skript verwendet. Dabei werden die vordefinierten Befehlssätze der Module geladen. Anschließend wird die Kameraauflösung angepasst. Danach wird innerhalb einer Schleife ein Foto geschossen. Da es beide Teilabschnitte des Versuches nahezu den selben Code verwenden wird in diesem Versuch die Schleife nur einmal durchlaufen, da nur ein Bild benötigt wird. Um keine Verluste durch eine Kompression der Bilddaten zu bekommen werden die Bilder im png-Format aufgenommen. Zuletzt wird auf der Konsole noch ein „done“ ausgegeben, um zu signalisieren, dass ein Bild geschossen wurde. Die folgende Abbildung 2.3 zeigt den Programmcode.

```

from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.resolution=(3280,2464)

camera.start_preview()
for i in range(1):
    sleep(3)
camera.capture('/home/pi/Pictures/image%s.png' % i,format='png')
camera.stop_preview
print("done")

```

Abbildung 2.3: Programmcode zum Schießen eines Bildes

An der Wand ist ein Testbild, mit einem Muster unterschiedlich dicker Striche, angebracht, wie in Abbildung 2.4 dargestellt. Dabei sind immer drei Striche mit gleicher Dicke und gleichem Abstand angebracht. Zudem sind die Striche Horizontal, sowie Vertikal angeordnet um beide Ausrichtungen testen zu können. In diesem Versuch werden die Vertikalen Strichmuster jedoch nicht Betrachtet.

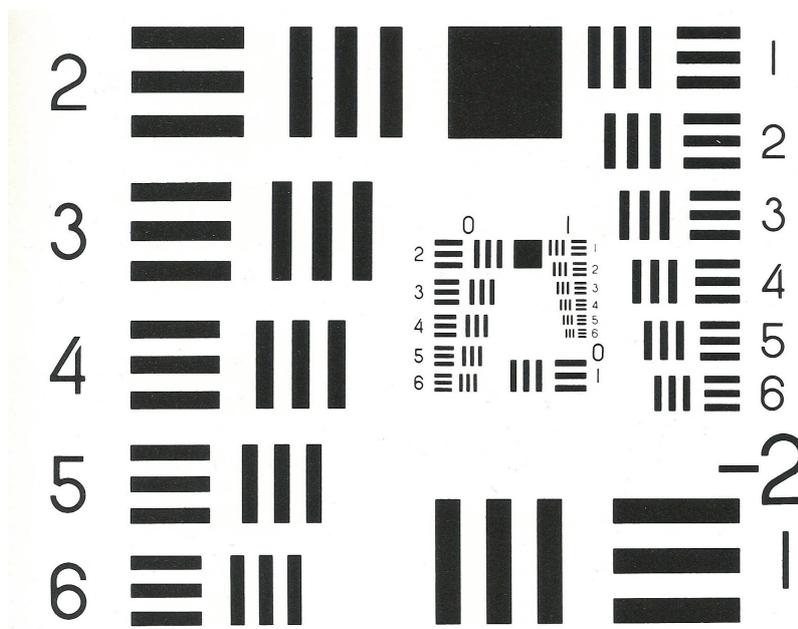


Abbildung 2.4: Testbild

2.2 Theoretische Grundlagen

In diesem Kapitel wird die theoretische Auflösung eines Kamerasystems, also die physikalische Fläche welche ein Pixel darstellt, berechnet. Somit können später die gemessenen Werte für die Auflösung in Relation gesetzt werden.

Zur Berechnung der Auflösung muss zunächst ermittelt werden, welche Fläche der Kamera-Sensor aufnimmt. In Kombination mit der Anzahl der Pixel, kann die Fläche pro Pixel und damit auch die Auflösung ermittelt werden.

Bei einer nach unten gerichteten Kamera, wie es beim Searchwing-Projekt der Fall ist, lässt sich die Fläche welche der Sensor aufspannt aus dem Öffnungswinkel des Kamerasystems und der Entfernung des Objektes zum Sensor ermitteln. Abbildung 2.5 stellt den Zusammenhang zwischen den Größen dar.

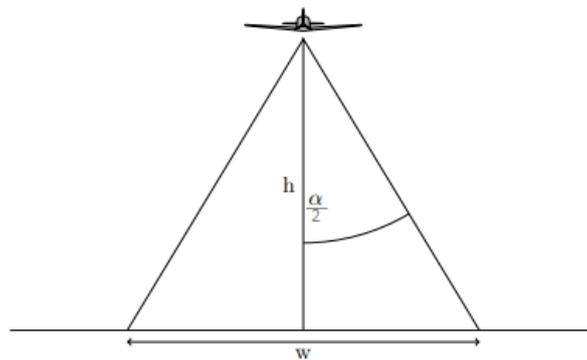


Abbildung 2.5: Breite am Boden bei einer nach unten gerichteter Kamera [8]

Da eine Kamera unterschiedliche vertikale (β) - bzw. horizontale Öffnungswinkel (α) besitzt, muss sowohl die horizontale- (w), als auch die vertikale Strecke (l) berechnet werden. Nach Abbildung 2.5 ergibt sich also für die Strecke „ w “, folgender Zusammenhang:

$$w = 2 \cdot \tan\left(\frac{\alpha}{2}\right) \cdot h \quad (2.1)$$

Analog hierzu kann die Strecke „ l “ ermittelt werden:

$$l = 2 \cdot \tan\left(\frac{\beta}{2}\right) \cdot h \quad (2.2)$$

Jetzt ist bekannt welche Fläche das Kamerasystem aufnimmt. Diese Fläche stellt die Kamera mit den ihr zur Verfügung stehenden Pixel dar. Um die Fläche eines Pixels, oder die Vertikale und horizontale Länge eines Pixels zu ermitteln, müssen die jeweiligen Strecken durch die Anzahl der Pixel in der dazugehörigen Dimension geteilt werden. Die Vertikale- bzw. Horizontale Auflösung lässt sich nun wie folgt berechnen:

$$g_w = \frac{w}{P_{hor}} = \frac{2 \cdot \tan\left(\frac{\alpha}{2}\right) \cdot h}{P_{hor}} \quad (2.3)$$

$$g_l = \frac{l}{P_{ver}} = \frac{2 \cdot \tan\left(\frac{\beta}{2}\right) \cdot h}{P_{ver}}$$

Tabelle 2.4 zeigt die für die Berechnung relevanten Parameter, welche in Abschnitt 2.1 beschrieben werden. Dabei ist anzumerken das bei der Highquality, sowie bei der Canon Kamera, ein Objektiv verwendet wird. Die Angaben zum Öffnungswinkel sind also vom verwendeten Objektiv abhängig.

Tabelle 2.4: Technische Spezifikationen der zu Untersuchenden Kamerasysteme

	Anzahl Pixel (P_{hor} x P_{ver})	Horizontaler Öffnungswinkel (α)	Vertikaler Öffnungswinkel (β)
Camera Module v2	3280 x 2464	62,2°	48,8°
HQ Camera	4056 x 3040	63,0°	49,6°
Canon EOS 600D	5184 x 3456	64,5°	45,5°

In Abbildung 2.6 wird die Horizontale Auflösung, also die Länge eines Pixels, in Abhängigkeit zur Distanz vom Sensor aufgezeichnet.

Es bildet sich ein linearer Verlauf. Je kleiner ein Pixel bei einer gewissen Entfernung ist, desto größer ist die Auflösung des jeweiligen Kamerasystems. Die Canon EOS 600D hat somit wie zu erwarten die beste und das V2 Kamera Modul die schlechteste Auflösung.

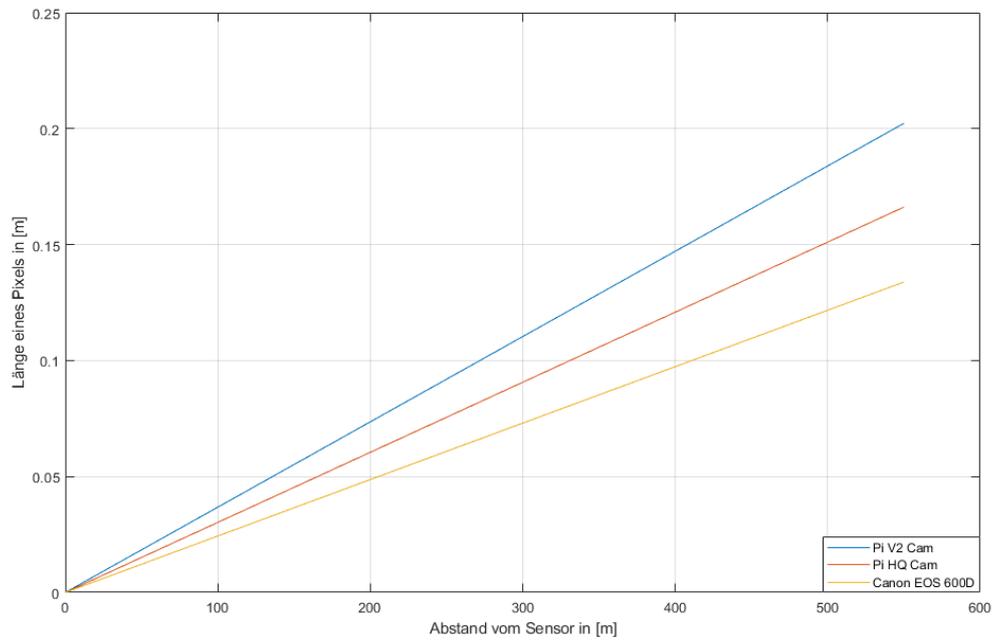


Abbildung 2.6: Berechnete Kennlinien der Kamerasysteme

Durch das Verhältnis des Öffnungswinkels zur Anzahl der Pixel ergibt sich für die Vertikale- und Horizontale Auflösung einer Kamera nahezu das selbe Ergebnis. Daher wird auf eine Betrachtung der Vertikalen Auflösung verzichtet.

Objekte welche real kleiner als die Länge eines Pixels sind können auf dem Bild nicht mehr erkannt werden. Bei einer Flughöhe der Drohne von 550 m können mit der verbauten V2 Kamera in der Theorie Objekte ab 20 cm Länge erkannt werden.

2.3 Versuchsdurchführung

Die Durchführung ist in zwei Teilabschnitte unterteilt. Im folgenden werden die beiden Abschnitte getrennt von einander beschrieben.

2.3.1 Messung im Bereich bis 20 m

Dieser Messversuch wird im Gang eines alten Betriebsgelände durchgeführt. Eine Stirnseite dient als Messwand, an welchem das Testbild befestigt ist. Um auch die Randbereiche in einem Bild testen zu können, sind an den Rändern eines Bildes weitere Testbilder angebracht. Damit kann untersucht werden, ob sich die Bildqualität in den Randbereichen ändert. Abbildung 2.7 zeigt die Messwand bei diesem Test.



Abbildung 2.7: Messwand für die Messreihe bis 20 m

Das Testbild ist wie in der Abbildung zu sehen im DIN A4 Format ausgedruckt. Die Messreihe beginnt bei einer Entfernung von zwei Metern und endet bei einer Entfernung von 20 m. Im Abstand von jeweils 2m wird ein Bild mit jeder Kamera aufgenommen.

2.3.2 Messung im Bereich bis 100 m

Um auch eine größere Entfernung zu Testen wird eine zweite Messreihe aufgenommen. Hierbei wird bei einer Entfernung von 10 Metern gestartet und mit 100 m aufgehört. Dabei wird alle 10 m ein Foto geschossen. Das Testbild wird hierbei auf einem größeren Format ausgedruckt, da die Abstände auf einem DIN A4 Ausdruck nicht ausreichend sind. Da auch die Strichstärken auf dem großen Ausdruck noch nicht ausreichen, um die Tests mit hohen Entfernungen durchzuführen, werden weitere eigens erstellte Ausdrücke verwendet. Die folgende Abbildung 2.8 zeigt die Messwand. Dieser Test wird auf einem Feldweg durchgeführt, da auf dem Betriebsgelände keine so großen Flächen zur Verfügung stehen.



Abbildung 2.8: Messwand für die Messreihe bis 100 m

2.4 Messergebnisse

Im folgendem wird beschrieben wie die aufgenommenen Bilder ausgewertet werden und die Ergebnisse der Versuche vorgestellt.

2.4.1 Auswerten der Ergebnisse

Zum besseren Verständnis der Ergebnisse, wird das Auswerten der Fotos anhand eines Bildes von der V2 Kamera in 20 m Entfernung erklärt (siehe Abbildung 2.9). Der Prozess ist bei den anderen Kamerasystemen identisch.



Abbildung 2.9: Versuchsfoto von der V2 Kamera in 20 m Entfernung

Um die Auflösung der Kamera (Länge eines Pixels) zu bestimmen, wird angeschaut welche Strichblöcke auf dem Testbild noch eindeutig zu erkennen sind. Hierzu muss im Versuchsfoto auf dem Bereich des Testbildes gezoomt werden.

In Abbildung 2.10 ist der vergrößerte Bereich des Testbildes 2.9 dargestellt. Hier ist zu sehen, dass beim Strichblock unten links (in rot Markiert) noch einzelne Schwarze Striche zu erkennen sind. Im nächst kleineren Strichblock, in Abbildung 2.10 oben rechts, verschwimmen die Striche bereits miteinander.

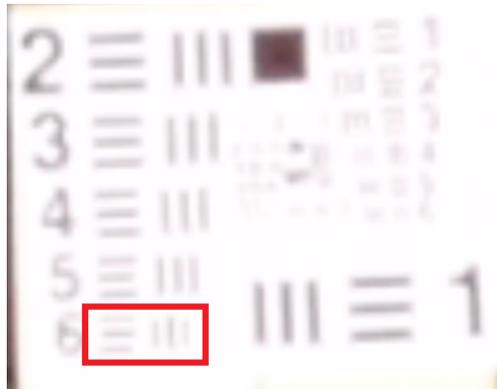


Abbildung 2.10: Zoom auf das Testbild aus Abbildung 2.9

Beim rot markierten Strichmuster handelt es sich also um die Striche mit den kleinsten noch zu erkennenden Abständen. Der Abstand wird auf dem Ausdruck des Testbildes mithilfe eines Messschiebers gemessen. Dieser entspricht hier 12,6 mm.

Bei einer Aufnahme mit der V2 Kamera in 20 m Entfernung lässt sich also eine Distanz von 12,6 mm gerade noch erkennen. Dieses Vorgehen wird für alle aufgenommenen Bilder wiederholt.

Bei dieser Art der Auswertung bildet sich ein systematischer Fehler aus. Durch die fest definierten Abständen des Strichmusters kann immer nur ermittelt werden zwischen welchen beiden Werten der tatsächliche kleinste noch erkennbare Abstand liegt. Zudem kommen Messunsicherheiten des Messschiebers und der Distanzmessung, sowie Ablesungenauigkeiten hinzu. Da diese nicht eindeutig identifizierbar sind, kann eine Messfehlereinschätzung nicht erfolgen.

2.4.2 Messung im Bereich bis 20 m

Es wurde mit den drei Kamerasystemen von 2 bis 20 Metern, in 2 m abständen, jeweils ein Foto aufgenommen. Diese wurden daraufhin wie beschrieben ausgewertet und in einem Koordinatensystem eingetragen (Blaue Kreise in den Abbildungen 2.11 bis 2.13). Aus diesen Messpunkten wurde jeweils eine Gerade approximiert und als blau gestrichelte Linie in den Diagrammen hinzugefügt. Zudem werden die jeweiligen theoretisch bestimmten Kennlinien der Kamerasysteme mit angegeben (rote Kennlinien), somit ist ein Vergleich zwischen Theorie und Praxis möglich.

Raspberrypi V2 Kamera

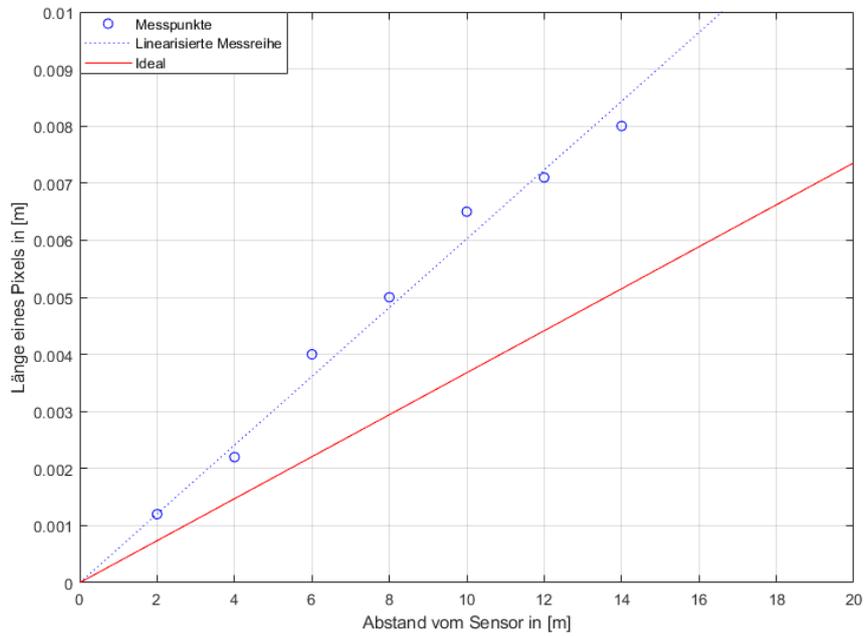


Abbildung 2.11: Messergebnisse der V2 Kamera von 2 bis 20 Metern

Hier ist Anzumerken, dass ab 16 m keine der Striche auf dem Testbild zu erkennen waren. Deshalb konnten keine Messpunkte für Entfernungen über 14 m gebildet werden. Um auch in höheren Distanzen Ergebnisse zu Erhalten, muss ein Testbild mit größeren Strichen bzw. Abständen verwendet werden.

Raspberrypi Hignquality Kamera

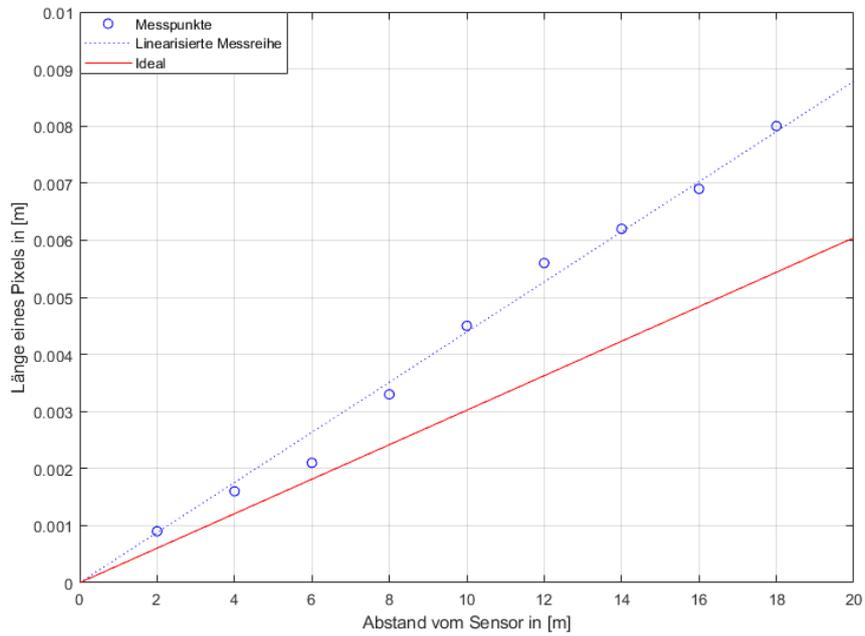


Abbildung 2.12: Messergebnisse der HQ Kamera von 2 bis 20 Metern

Auch hier reichte das verwendete Testbild nicht aus, in 20 m Entfernung konnten keine individuellen Striche erkannt werden. Somit fehlt hier der Messpunkt in 20 m Entfernung.

Canon EOS 600D

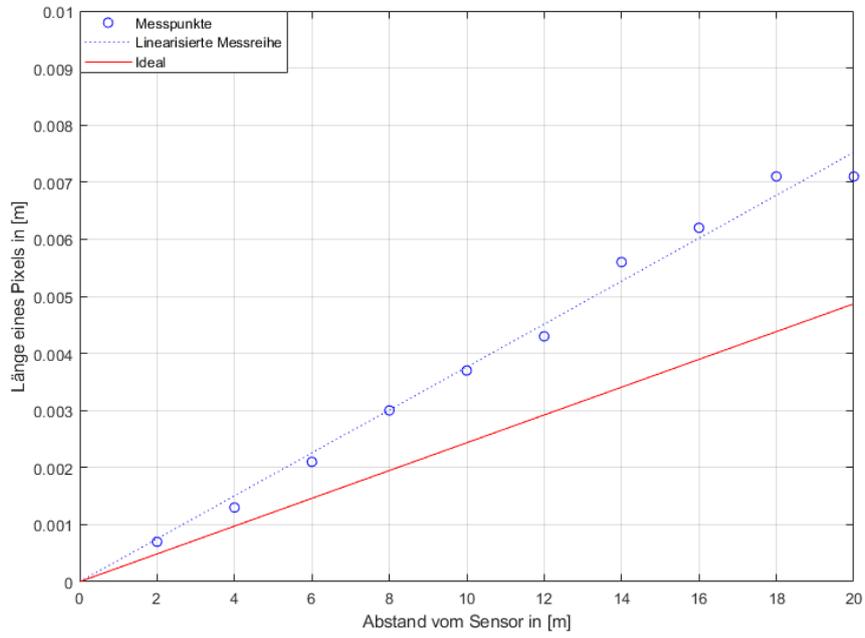


Abbildung 2.13: Messergebnisse der Canon Kamera von 2 bis 20 Metern

Vergleich der Kamerasysteme

Um einen Vergleich zwischen den Kamerasystemen zu ermöglichen, werden die jeweils linearisierten Messreihen in einem Diagramm angegeben.

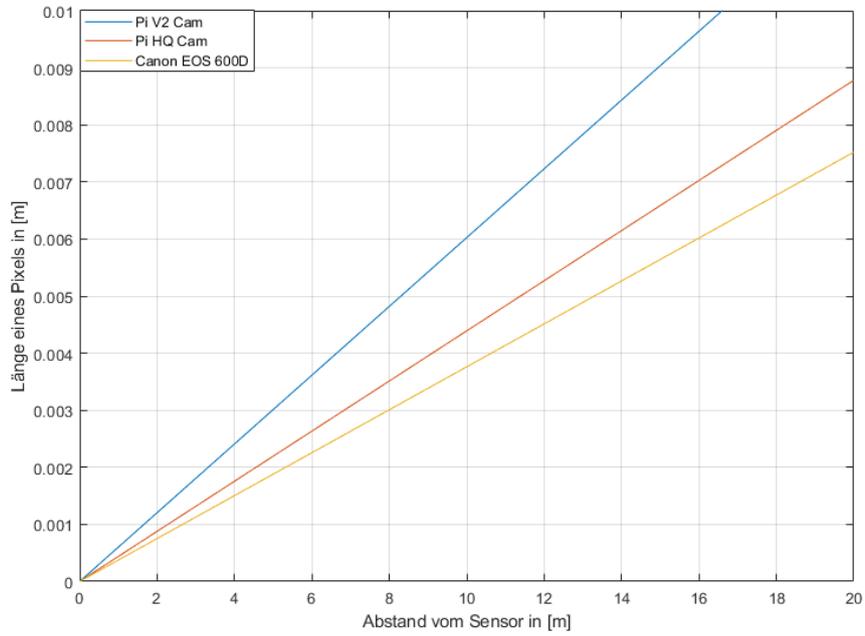


Abbildung 2.14: Vergleich der Kamerasysteme von 2 bis 20 Metern

2.4.3 Messung im Bereich bis 100 m

Die Auswertung und Darstellung der Ergebnisse erfolgt wie bei der Messung bis 20 m.

Raspberry V2 Kamera

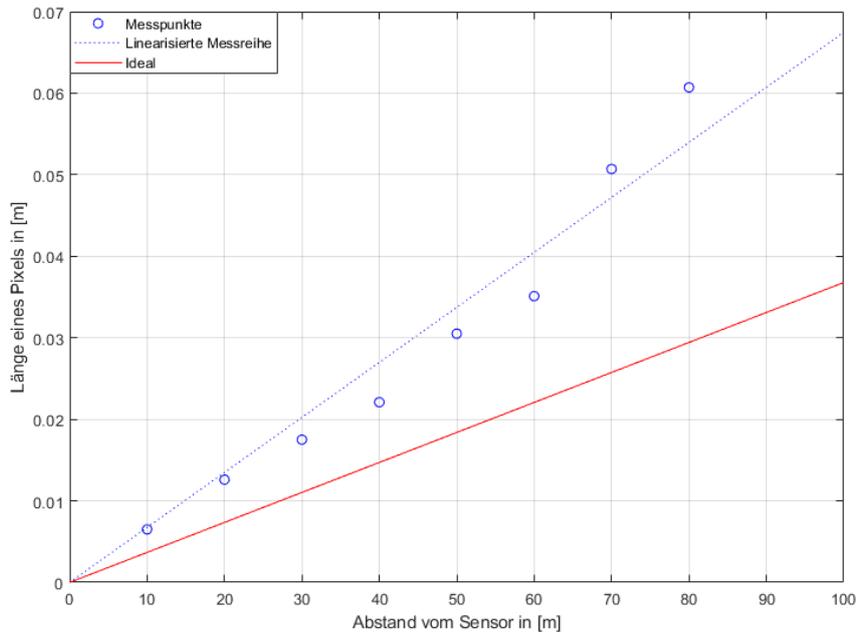


Abbildung 2.15: Messergebnisse der V2 Kamera von 10 bis 100 Metern

Auch hier waren ab 90 m keine Striche mehr zu sehen, weshalb die Messpunkte fehlen.

Raspberrypi Highquality Kamera

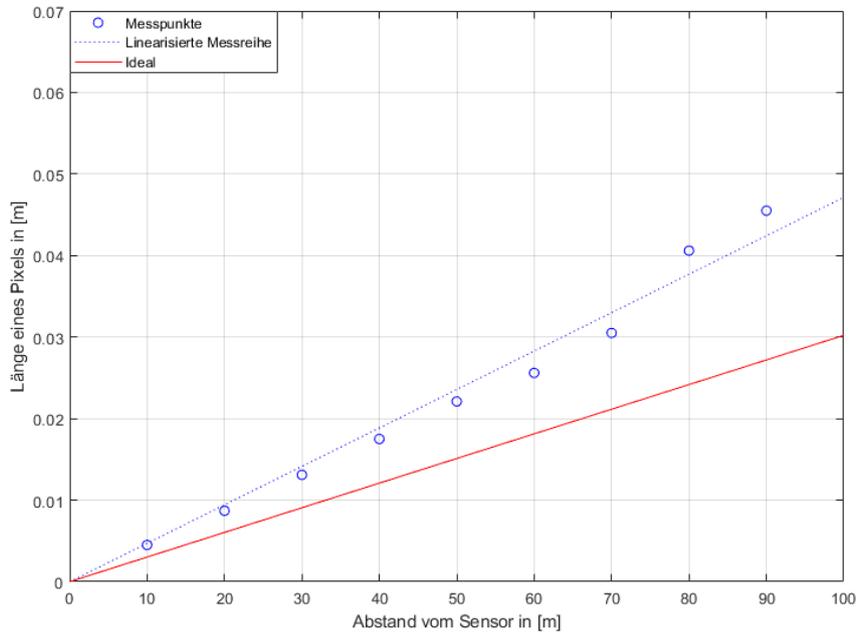


Abbildung 2.16: Messergebnisse der HQ Kamera von 10 bis 100 Metern

Vergleich der Kamerasysteme

Nun können wieder die linearisierten Messreihen der Kamerasysteme miteinander verglichen werden.

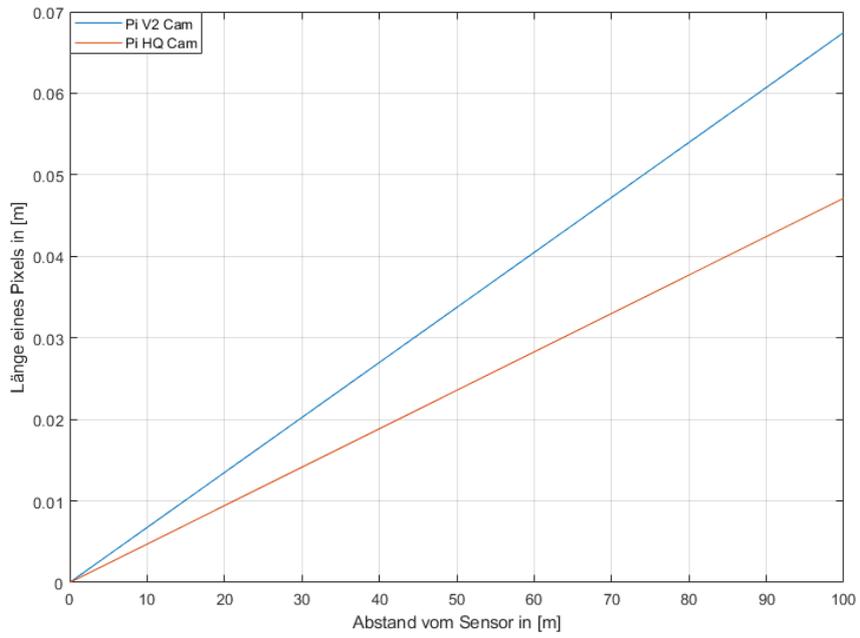


Abbildung 2.17: Vergleich der Kamerasysteme von 10 bis 100 Metern

2.5 Diskussion der Ergebnisse

Bei der Messung im Bereich bis 20 m spiegeln die ermittelten Messwerte nicht die Erwarteten wieder. Zu Erwarten waren annähernd die theoretisch errechneten Werte wie in Abschnitt 2.2 beschrieben. Jedoch sind die gemessenen Werte bei allen Kameras ca. 60 % über den Erwartungen. Da bei allen die Öffnungswinkel ähnlich sind kann jedoch erkannt werden, dass eine höhere Pixelanzahl zu einer kleineren Länge pro Pixel führt. Dieser Zusammenhang ist auch schon aus den idealen Kennlinien ersichtlich. Daher lässt sich sagen, dass die V2 von Raspberry pi die schlechteste Kamera in diesem Test ist. Das HQ Kameramodul liegt in der Mitte und die Canon EOS 600D Kamera schneidet am besten ab. Warum genau die Messwerte so stark von den idealen Werten abweichen konnte aus zeitlichen Gründen und fehlendem Wissen im Bereich der Optoelektronik nicht vollständig beantwortet werden. Jedoch haben Recherchen ergeben, dass dies zum Beispiel am verwendeten Bayern-Pattern der Sensoren oder dem Interpolationsverfahren liegen kann.

Die Messung im Bereich bis 100 m dient hauptsächlich zu Verifizierung des linearen Verhaltens der Messergebnisse. Genau dieses Verhalten konnte nachgewiesen werden. Damit ist davon auszugehen, dass das Verhältnis zwischen Abstand und Länge eines Pixels über den gesamten Bereich linear verläuft. Für diesen Versuch stand die Canon EOS 600D Kamera nicht zur Verfügung. Jedoch zeigen die Messwerte der beiden anderen Kameramodule, dass auch das Verhalten dieser Kamera linear ist.

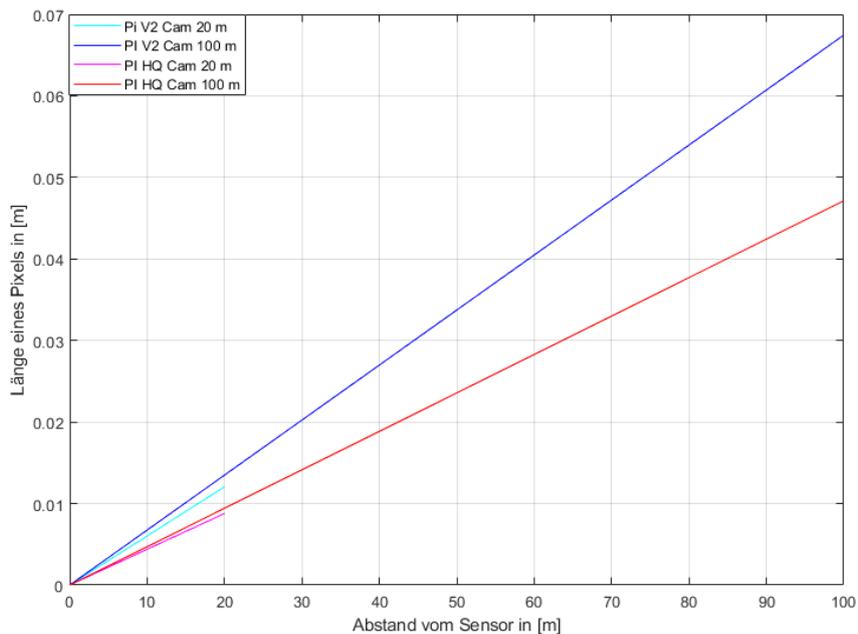


Abbildung 2.18: Vergleich der zwei Messreihen

3 Schlussfolgerung und Fazit

Im ersten Teil der Arbeit wurde der Stromverbrauch des Raspberry Pi unter verschiedenen Rahmenbedingungen berechnet. Die Messungen des IDLE-Modus haben gezeigt, dass die Messungen trotz gleicher Rahmenbedingungen sehr volatil sind, da der Raspberry Pi selbständig Prozesse startet und beendet auf die man von außen keinen Zugriff hat. Dadurch ergibt sich, dass bei gleichen Rahmenbedingungen die Messwerte um über 10 % schwanken können. Deshalb ist es schwierig, präzise anzugeben, wie viel Mehrenergie beispielsweise das Anschalten des Wlans benötigt. Dennoch liegen die Messwerte in einer ähnlichen Größenordnung, wie Messdaten, die im Internet zu finden sind. Bei den Messungen des Stromverbrauchs beim Schießen von Bildern konnte gezeigt werden, dass das Abspeichern im JPEG-Format deutlich weniger Energie benötigt, als das Abspeichern als PNG-Format. Des Weiteren führt vor allem das Initialisieren der Kamera zu längeren Programmzeiten. Ist diese erst einmal aktiviert, können schnell mehrere Bilder hintereinander aufgenommen werden. Mit Hilfe von extra Pausen im Programm konnte zudem zugeordnet werden, welcher Teil des Programms für welchen Energiemehrverbrauch verantwortlich ist.

Im zweiten Teil der Arbeit zeigt sich, je mehr Pixel eine Kamera besitzt, desto kleiner ist die Länge eines Pixels. Dieser Vergleich kann jedoch nur gemacht werden, wenn die Öffnungswinkel vergleichbar sind. Zudem lässt sich erkennen, dass die rechnerisch Ermittelten Längen eines Pixels nicht erreicht werden. Die genauere Betrachtung der Ursachen hierfür konnten jedoch in dieser Arbeit aus zeitlichen Gründen nicht gemacht werden. Kleinere Recherchen haben ergeben, dass zum einen das verwendete Bayern-Pattern der Kamerasensoren, sowie das Interpolationsverfahren eine deutliche Auswirkung auf die Pixel haben. Für eine genaue Aussage müssen weitere Untersuchungen gemacht werden.

Abbildungsverzeichnis

1.1	Versuchsaufbau mit Raspberry Pi und Shunt-Widerstand an Labornetzteil . . .	3
1.2	Verschaltung des Versuchsaufbaus als Schaltplan	4
1.3	Exemplarischer Programmcode zum Schießen von drei Bildern in Reihe mit anschließender Pause von einer Sekunde	6
1.4	Stromverbrauch bei eingeschaltetem Wlan ohne Netzverbindung	7
1.5	Anstieg des Stromverbrauchs während des Schießens eines einzelnen Fotos . .	8
1.6	Anstieg des Stromverbrauchs während des Schießens von fünf Fotos direkt hintereinander	9
1.7	Stromverlauf bei Abspeicherung eines einzelnen Bildes als png-Datei	10
1.8	Messung von fünf Fotos hintereinander mit zusätzlicher Pause nach dem Einschalten der Kamera von 3 s	11
1.9	Messung eines Fotos mit zusätzlicher Pause nach dem Einschalten der Kamera von 3 s	11
1.10	Stromverlauf beim Schießen von drei Bildern in Reihe mit einer Pause von einer Sekunde nach jeder Aufnahme	12
2.1	schematischer Messaufbau	17
2.2	Messstand	18
2.3	Programmcode zum Schießen eines Bildes	19
2.4	Testbild	19
2.5	Breite am Boden bei einer nach unten gerichteter Kamera [8]	20
2.6	Berechnete Kennlinien der Kamerasysteme	22
2.7	Messwand für die Messreihe bis 20 m	23
2.8	Messwand für die Messreihe bis 100 m	24
2.9	Versuchsfoto von der V2 Kamera in 20 m Entfernung	25
2.10	Zoom auf das Testbild aus Abbildung 2.9	26
2.11	Messergebnisse der V2 Kamera von 2 bis 20 Metern	27
2.12	Messergebnisse der HQ Kamera von 2 bis 20 Metern	28
2.13	Messergebnisse der Canon Kamera von 2 bis 20 Metern	29
2.14	Vergleich der Kamerasysteme von 2 bis 20 Metern	30
2.15	Messergebnisse der V2 Kamera von 10 bis 100 Metern	31
2.16	Messergebnisse der HQ Kamera von 10 bis 100 Metern	32
2.17	Vergleich der Kamerasysteme von 10 bis 100 Metern	33
2.18	Vergleich der zwei Messreihen	34

Tabellenverzeichnis

1.1	Überblick über wichtigste Spezifikationen des Raspberry Pi Zero W [1]	2
1.2	Mittlerer Strom und mittlere Leistung der Messung des IDLE-Modus	7
1.3	Mittlerer Strom und mittlere Leistung beim Schießen von Fotos	8
1.4	Energiekosten für das Schießen eines Fotos bzw. mehrerer Fotos	13
2.1	Übersicht über wichtigste Spezifikationen des V2 Kamera Moduls[4]	16
2.2	Übersicht über wichtigste Spezifikationen des HQ Kamera Moduls [4, 5] . . .	16
2.3	Übersicht über wichtigste Spezifikationen der Canon Kamera [6, 7]	17
2.4	Technische Spezifikationen der zu Untersuchenden Kamerasysteme	21

Literaturverzeichnis

- [1] GIGA-Redaktion. <https://www.giga.de/hardware/raspberry-pi-zero-w>, 2017.
- [2] Manuel. <https://www.datenreise.de/raspberry-pi-stromversorgung-netzteil-empfehlung>, 2018.
- [3] JUERGEN. Stromverbrauch des raspberry pi zero w (mit wlan und bluetooth) gemessen (<https://www.bitblokes.de/stromverbrauch-des-raspberry-pi-zero-w-mit-wlan-und-bluetooth-gemessen>, 2020).
- [4] Camera module - raspberry pi documentation, 29.12.2020.
- [5] pi3g. 6mm weitwinkelobjektiv für hq kamera (6mm wide angle lens for hq camera), 29.12.2020.
- [6] Canon eos 600d - eos digital slr and compact system cameras - canon deutschland, 29.12.2020.
- [7] Canon ef-s 18-55mm f/3.5-5.6 is ii -specification - lenses - camera & photo lenses - canon deutschland, 29.12.2020.
- [8] searchwing-drohne-allgemein [digitale elektronik und technische informatik], 03.01.2021.