Lukas Pajak

Automatisierung einer Meteorkamera

lukas.pajak@gmx.de

Jugend forscht 2019 - Nordrhein-Westfalen - Geo- und Raumwissenschaften

Betreuer: Bernd Koch und Michael Winkhaus

Carl-Fuhlrott-Gymnasium Wuppertal

Inhaltsverzeichnis

1. Einleitung ... 2
1.1. Begriffserklärung ... 2
1.2. Das Europäische Feuerkugelnetz ... 3
2. Die Meteorkamera des Carl-Fuhlrott-Gymnasiums ... 3
2.1. Inneres Gehäuse ... 4
2.2. Hardware ... 5
2.3. Der Shutter und die dazugehörige Steuerung ... 6
2.3.1. Kalibrierung der Geschwindigkeit des Shutters ... 7
2.4. Software ... 9
2.4.1. Sensorprogramm ... 10
2.4.2. Ermittlungsprogramm ... 10
2.4.3. Dauerprogramm ... 11
3. Auswertung ... 11
4. Zusammenfassung ... 12
5. Danksagung ... 12
Quellen ... 13
Anhang 1: Lichtsensorprogramm in Python3 ... 14
Anhang 2: Ermittlungsprogramm in Bash ... 15

1. Einleitung

Eine Meteorkamera nimmt langzeitbelichtete Bilder des Nachthimmels auf, wobei das Bild regelmäßig unterbrochen wird. Dank dieser Unterbrechungen im Bild und noch mit Hilfe von Aufnahmen aus anderen Perspektiven lässt sich die Flugkurve von Himmelskörpern berechnen.

Mein Ziel ist es ein automatisches System zu entwickeln und zu bauen, welches jede Nacht Aufnahmen des Nachthimmels macht.

Letztes Jahr habe ich Rahmen des Projektkurses Astronomie das System angefangen zu entwickeln und zu bauen. Am Ende wurde jede Nacht während eines definierten Zeitintervalls Bilder mit einer festen Belichtungszeit von 15 Minuten aufgenommen und auf den lokalen Schulserver gespeichert. Zwar konnte man Belichtung und Uhrzeit auf die Monate anpassen, aber da sich die Lichtverhältnisse nur schlecht vorhersagen lassen und es am Ende sowohl unter-, als auch überbelichtete Bilder gab, habe ich mich dazu entschieden ein Lichtsensor in das System zu integrieren. Dieser erlaubt dynamische Start -und Stoppzeitpunkte und eine ideale Belichtungszeit.

1.1. Begriffserklärung

Ein Körper, der (noch) nicht die Atmosphäre betreten hat und kleiner als 1 km ist, nennt man **Meteoroid**. Sehr selten kommen auch **Earthgrazer** vor, Meteore, die nur von kurzer Dauer die Erdatmosphäre streifen und wieder ins Weltall fliegen. Ein **Meteor** ist eine Leuchterscheinung in der Erdatmosphäre, wobei man hier auch von einer **Sternschnuppe** spricht, solange die höchstens nur ein paar Zentimetern groß ist. Sternschnuppen zerfallen immer fast völlig, sind aber mit dem bloßen Auge sichtbar. Falls der Himmelskörper über 10cm Durchmesser erweist, bezeichnet man den als **Feuerkugel**. Diese sind auch als Boliden bekannt und sind heller als -4 mag[1]. Wenn es eine Feuerkugel bis auf den Boden schafft, nennt man die Überreste **Meteorit**.

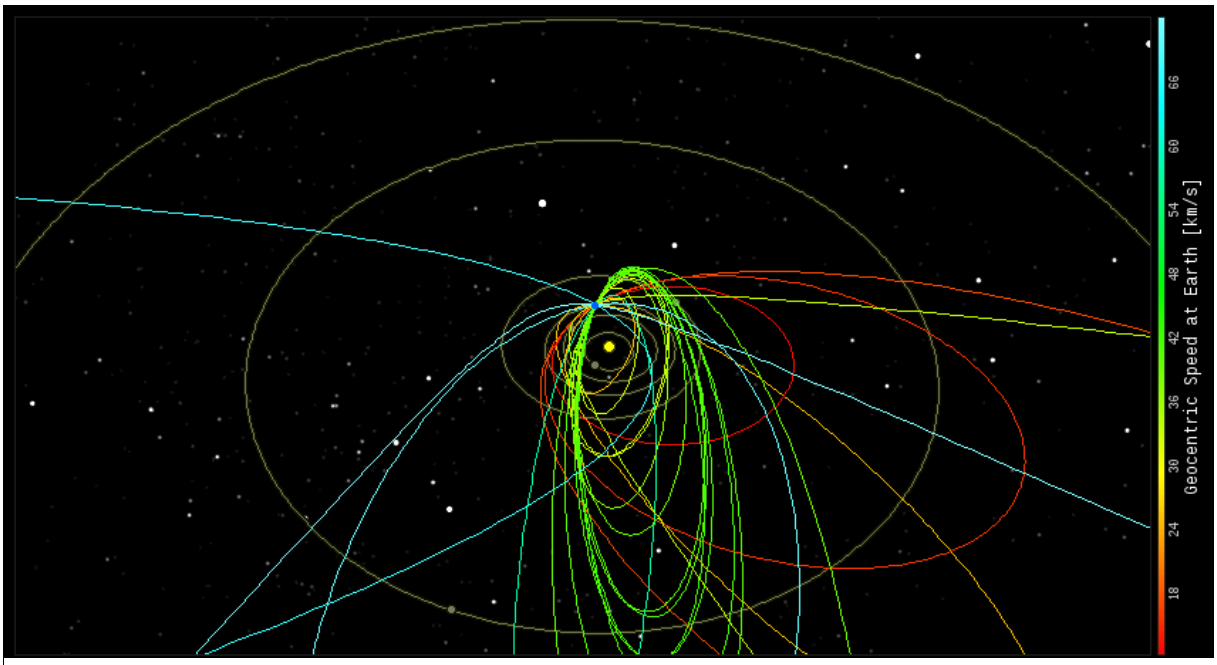

Abbildung 1 Am 5. Januar 2019 wurden 30 Meteore vom automatischem „NASA all-sky cameras“ System erfasst. Die Farbe der Bahnen ist relativ zur geometrischen Geschwindigkeit der einzelnen Himmelskörper.
Quelle: www.spaceweather.com/images2019/05jan19/orbits.gif

[1] Die auf der Erde scheinbare Lichtintensität. Ein niedrigerer Wert bedeutet eine hellere Erscheinung.

Prinzipiell laufen Meteoroiden auf elliptischen bis hyperbolischen Bahnen um die Sonne. Falls ein Himmelskörper eine elliptische Bahn annimmt, bleibt dieser auch im System. Anders ist es bei Hyperbel- oder Parabelbahnen, dort verlassen die Meteoriden das Sonnensystem wieder. Das betrifft aber gerade mal weniger als 1% der Meteoroiden.

1.2. Das Europäische Feuerkugelnetz

Europa besitzt leider kein vollautomatisches System wie die USA, jedoch gibt es hier ein Verbund aus vielen leidenschaftlichen Astronomen, welche ehrenamtlich einzelne Stationen betreuen, in denen regelmäßig Aufnahmen gemacht werden.

Den Verbund selbst gibt es schon seit den siebziger Jahren und 2002 gab es die erste erfolgreiche Suche auf Basis von fotografischen Beobachtungen in ganz Deutschland. Alle drei Fundstücke gehörten mal zu einem großen Meteoriten, dessen Hauptmasse noch nicht gefunden wurde, wobei diese auf bis zu 10 Kilogramm geschätzt wird. Die gefundenen Meteoriten werden nach dem berühmten Schloss, welches sich in der Nähe des Fundortes befindet, benannt: Neuschwanstein. Es war ein großer Erfolg und ging sogar damals durch die Presse.

In den letzten 9 Jahren wurden ganze 365 Feuerkugeln von dem Verbund aufgenommen und registriert.

2. Die Meteorkamera des Carl-Fuhlrott-Gymnasiums

Grundsätzlich wird diese Meteorkamera auf einen Parabolspiegel gerichtet, sodass man mit einer Aufnahme den ganzen Nachthimmel im Blick hat. Mit Hilfe von Aufnahmen derselben Feuerkugel von einem anderen Standort aus kann die Höhe und die Geschwindigkeit des in die Erdatmosphäre eintretenden Meteoriden berechnet werden. Dazu befindet sich ein Shutter vor dem Objektiv.

Der Spiegel wird beheizt, damit bei allen Wetterbedingungen Aufnahmen möglich sind.

Das gesamte System befindet sich auf dem Dach der Schule.

Abbildung 3 Die Meteorkamera

Abbildung 4 Der beheizte Parabolspiegel

2.1. Inneres Gehäuse

Da die Box vom letzten Jahr, in der alle Komponenten gelegt werden, nicht den benötigten Platz bietet, habe ich eine neue, höhere Box aus Spanplatten gebaut, welche auch eine Fixierung der Teile durch Schrauben erlaubt um Kurzschlüsse und andere Dilemmata zu vermeiden. Ein großes Loch für die Kamera mit noch größeren Karosseriescheiben erlauben eine genaue Justierung, damit möglichst frontal auf die Spiegelmitte fotografiert wird.

Um das Holz vor Luftfeuchtigkeit zu schützen, wurde es mit einer Holzschutzlasur lackiert.

Abbildung 5 Die fertige Box trocknet vom Lack

Abbildung 6 Die meisten Komponenten sind noch nicht eingebaut.

Abbildung 7 Eingebauter Zustand, ohne Deckel

2.2. Hardware

Ich habe mich für eine schon vorhandene Digitalkamera entschieden, da diese sowohl mit der Software funktioniert als auch akzeptable Bildqualität bietet. Es handelt sich um die Canon EOS E500D, welche einen CMOS-Sensor mit 15,1 Megapixeln verbaut hat. Als Objektiv reicht das Kitobjektiv vollkommen. Bilder können außerdem im RAW Format und im JPEG Format gespeichert werden, was einem die Möglichkeit gibt, die Bilder sehr effektiv zu bearbeiten, aber auch schnell alle Bilder mal durchzuschauen.

Diese Kamera wird von einem Einplatinencomputer gesteuert, genauer mit dem Raspberry Pi 3. Der bietet genug Leistung für die Aufgaben und USB Anschlüsse, wodurch die Steuerung sehr einfach ist. Ein Netzwerkkabel wurde schon verlegt, ursprünglich mit einem USB-Adapter ausgestattet, nun verwende ich es als normales Netzwerkkabel, womit der Einplatinencomputer Zugriff auf das Internet und auf das lokale Schulnetzwerk hat.

Als Spannungsquelle wird ein Netzteil der Firma Mean Well verwendet. Es liefert konstante 5 Volt und kann bis zu 12 Ampere liefern und bietet eine hohe Zuverlässigkeit. Der Raspberry Pi und die Motorsteuerung sind direkt an das Netzteil angeschlossen. Da die Kamera eine Spannung eines Zweizellen Lithium-Ionen-Akkus mit ca. 7,2 Volt erwartet, wird die Spannung mit Hilfe eines Step-Up Spannungsregler der Firma Pololu hochgeregelt. Dieser Regler ist an einen Akku-Dummy geschaltet, welcher wie ein herkömmlicher Akku in die Kamera positioniert wird.

Die einzelnen Geräte habe ich mit SiF[2] Litzen verbunden, da diese flexibel ist und für die Verkabelung in Schaltanlagen und Verteilen geeignet ist.

Um Kabel miteinander zu verbinden werden Verbindungsklemmen der Firma Wago (Bsp.: Abbildung 8) verwendet. Diese bieten eine hohe Flexibilität, die bei den ersten Testläufen schon sehr oft ihren Vorteil ausspielen konnten.

Abbildung 8 Eine Verbindungsklemme der Firma Wago
Quelle: www.conrad.de/de/verbindungsklemme-flexibel-008-4-mm-starr-008-25-mm-polzahl-3-wago-222-413-15-st-grau-orange-730187.html

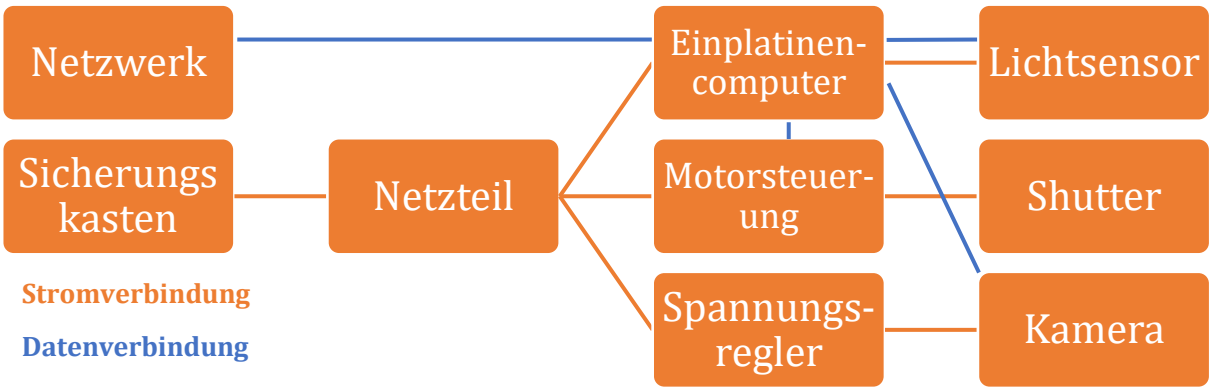

Abbildung 9 Schaubild zur Veranschaulichung der Verbindungen unter den Geräten.

[2] Silikon-Aderleitung

Abbildung 12 Der gelbe Chip ist der eigentliche Sensor

Abbildung 11 Die wettergeschützte Bohrung

Um die Lichtintensität des Himmels zu messen wird ein Breakoutboard der Firma Adafruit mit dem TSL2591 Lichtsensor (Abbildung 10) verwendet. Dieser bietet einen großen Messbereich und mit Hilfe des separaten Infrarotsensor, lassen sich die Wellenlängen herausfiltern, welches die Kamera aufnimmt, weil die Intensität nicht durch Infrarotlicht erhöht wird, wie es oft bei Helligkeitssensoren der Fall ist.

Abbildung 10 Adafruit TSL2591

Quelle: www.adafruit.com/product/1980

2.3. Der Shutter und die dazugehörige Steuerung

Der alte Shutter funktioniert noch einwandfrei, wobei ich eine neue Steuerung für diesen gebaut habe. Der Motor ist für 12V ausgelegt, jedoch verwende ich 5V, da ich dann nur ein Netzgerät für das gesamte System brauche und außerdem verbraucht der Motor dann weniger Strom bei der gleichen Drehzahl, weil weniger Strom bei dem Einstell-Widerstand in Hitze umgewandelt wird.

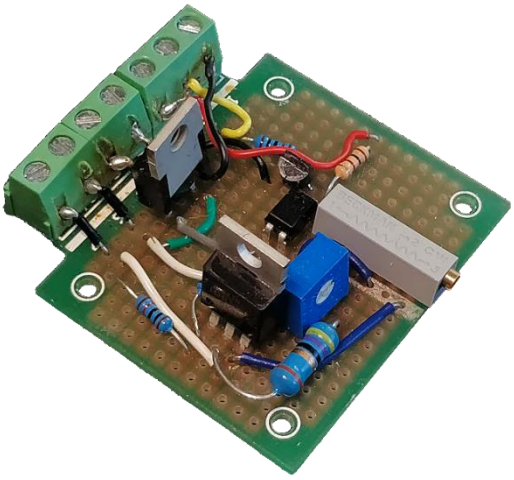

Abbildung 13 Fertiggebaute Steuerung

Die Steuerung ermöglicht erstens Mithilfe von zwei Potentiometern, eine Grob- und eine Feinjustierung, die Stromlimitierung und damit die Einstellung der Geschwindigkeit des Shutters und zweitens einen reibungslosen Start des Shutters. Da der Motor manchmal Probleme hat mit geringem Strom zu starten, lässt sich der Potentiometer überbrücken, wodurch mehr Strom fließt. Das Timing übernimmt wieder der Raspberry Pi, dessen Anschlüsse auf der Platine mit Hilfe eines Optokopplers galvanisch getrennt ist.

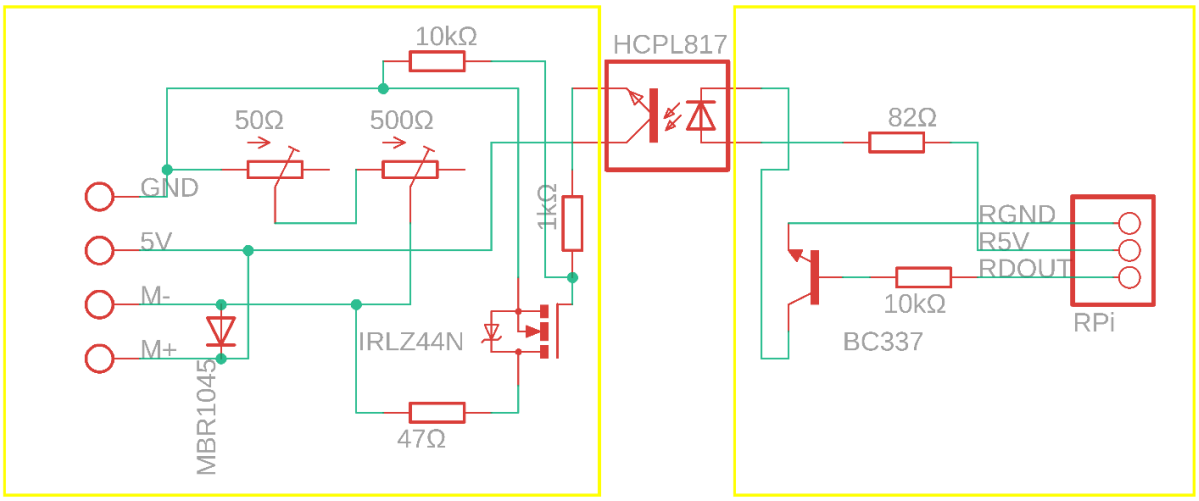

Abbildung 14 Schaltplan der Motorsteuerung

Letztes Jahr war nur eine grobe Einstellung der Frequenz möglich, weshalb nur ein unzureichendes Ergebnis möglich war. Da sich ohmsche Widerstände in Reihe einfach addieren, war eine Verbessrung mit einem feineren einstellbaren Widerstand möglich.

2.3.1. Kalibrierung der Geschwindigkeit des Shutters

Da nun eine Feinjustierung möglich ist, kalibriere ich den Motor nochmal.

Durch ein optisches Frequenzmessgerät der Firma SKF lässt sich die Umdrehungen pro Minute des Motors messen, indem das Gerät mit einem Laser (Abbildung 15) auf den Reflektor des Shutters (Abbildung 16) strahlt und die Reflektionen misst. Zusätzlich blinkt das Gerät mit der gemessenen Zeit gegen den Reflektor um die Präzision zu erhöhen. Nach möglichst genaustem einstellen, pendelte sich der Wert auf 899,5 Umdrehungen pro Minute ein. Das entspricht ca. 14,99Hz, knapp unter dem Standard von 15Hz des europäischem Feuerkugelnetzes. Die Bildunterbrechung erfolgt ca. 30-mal pro Sekunde, da der Shutter zwei Rotorblätter besitzt. Nun werden schnell durchs Bild wandernde Objekte zerlegt, wie in Abbildung 17, wobei die Abstände zwischen den einzelnen Bildpunkten eine Auskunft über die Geschwindigkeit geben können.

Abbildung 15 Modul bestehend aus einer Laserdiode und Sensor

Abbildung 16 Reflektor am Shutter

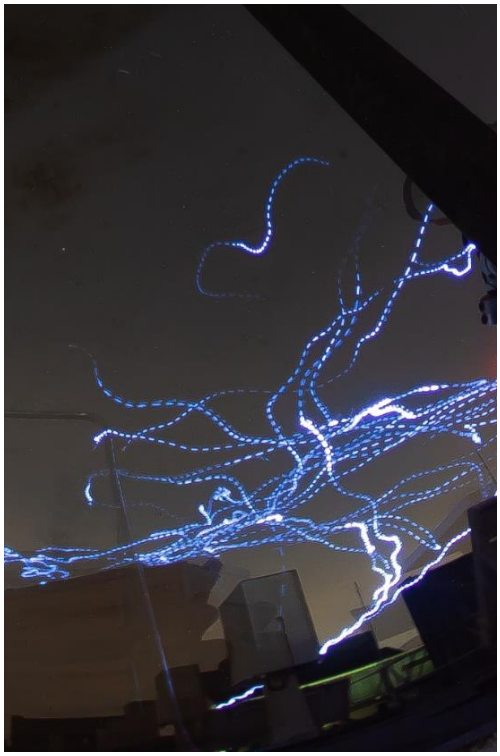

Abbildung 17 Aufnahme mit zerhackten Lichtspuren. Es entstand an einem Abend, an dem an der Station neben der Meteorkamera gearbeitet wurde.

Abbildung 18 Messsaufbau

2.4. Software

Die Wahl des Betriebssystems des Einplatinencomputers ist einfach, da es ein extra für den Raspberry Pi angepasstes Debian Linux namens Raspbian Stretch Lite existiert. Es ist schon komplett für den Raspberry Pi eingerichtet und es müssen nur noch Kleinigkeiten, wie der SSH-Server[3], konfiguriert werden. Durch diesen hat man vollen Zugriff auf das System, ohne mit einem Bildschirm und Tastatur am Einplatinencomputer zu sein. Man muss sich muss lediglich im selben Netzwerk befinden. Die Lite Version verzichtet auf die Benutzeroberfläche und bietet nur einen Kommandozeilen-Zugriff. Das spart Rechenleistung und da kein Monitor angeschlossen ist, wäre ein Desktop überflüssig.

Da zum Abgleichen mit anderen Meteorkameras die Uhrzeit eine große Rolle spielt, installierte ich die Software ntpd, welche die Zeit mit Servern synchronisiert, wobei das Programm konstant Abweichungen beachtet und gegebenenfalls die Zeit korrigiert. Außerdem wählt das Programm automatisch die beste Zeitquelle aus. Um Abgleichungen mit Onlinedatenbanken zu vereinfachen verwendet das gesamte System UT-Zeit.

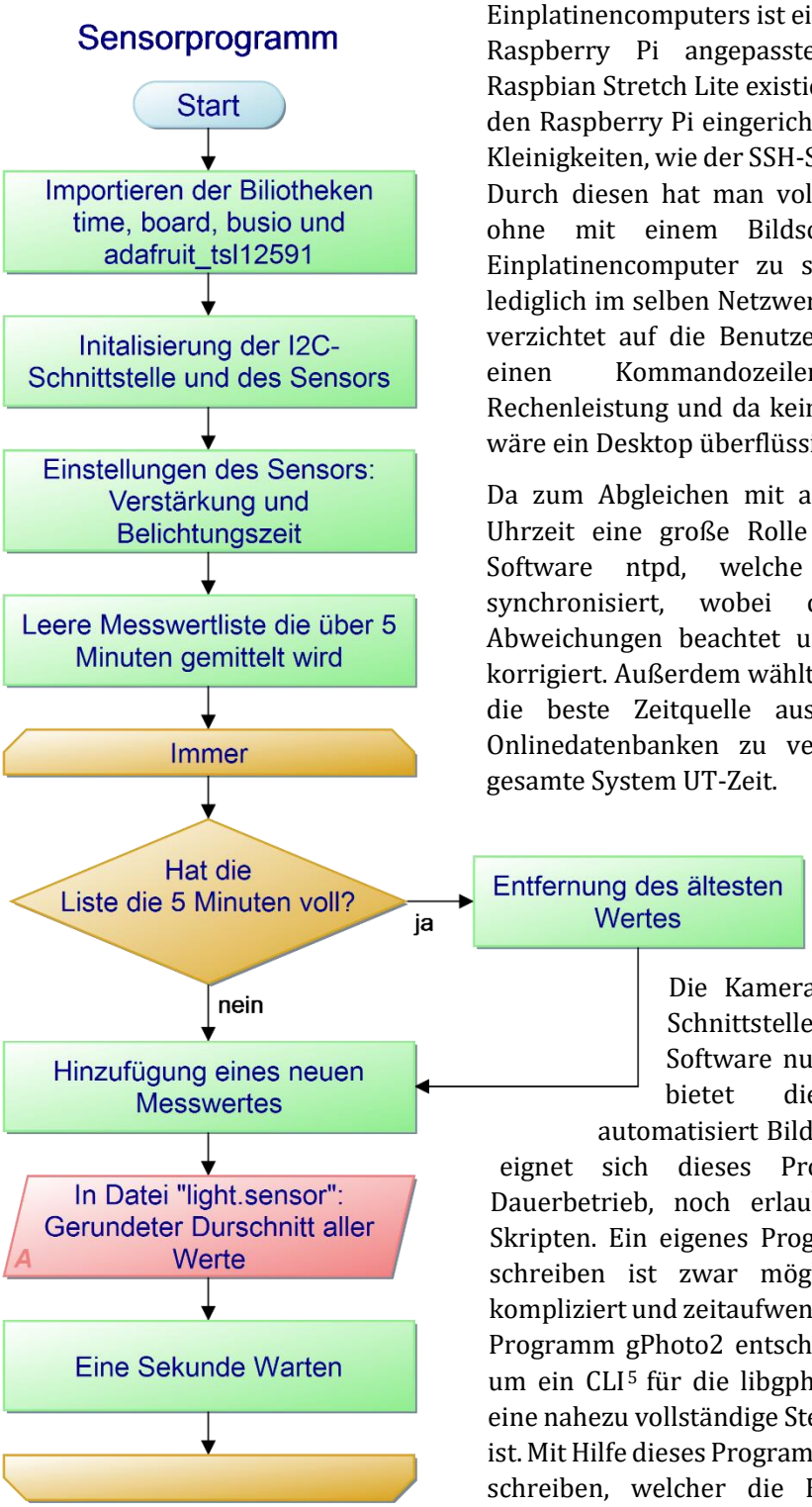

Abbildung 19 Programmablaufplan für die Lichtmessung

Die Kamera lässt sich über die USB-Schnittstelle steuern, wobei es offizielle Software nur für Windows gibt. Zudem bietet diese keine Möglichkeit automatisiert Bilder aufzunehmen. Außerdem eignet sich dieses Programm weder für den Dauerbetrieb, noch erlaubt es die Benutzung von Skripten. Ein eigenes Programm mit Canons SDK[4] zu schreiben ist zwar möglich, jedoch ist das sehr kompliziert und zeitaufwendig, weshalb ich mich für das Programm gPhoto2 entschieden habe. Es handelt sich um ein CLI[5] für die libgphoto2, ein Programm womit eine nahezu vollständige Steuerung der Kamera möglich ist. Mit Hilfe dieses Programmes kann ich ein Bash-Script schreiben, welcher die Kamera steuert, ohne ein

[3] Secure Shell – Aktuell einer der sichersten und gängigsten Möglichkeiten eine Datenverbindung zu einem Betriebssystem aufzubauen
[4] Software Development Kit – Programmcode zum entwickeln eigener Anwendungen
[5] Command Line Interface – Kommandozeilen Schnittstelle

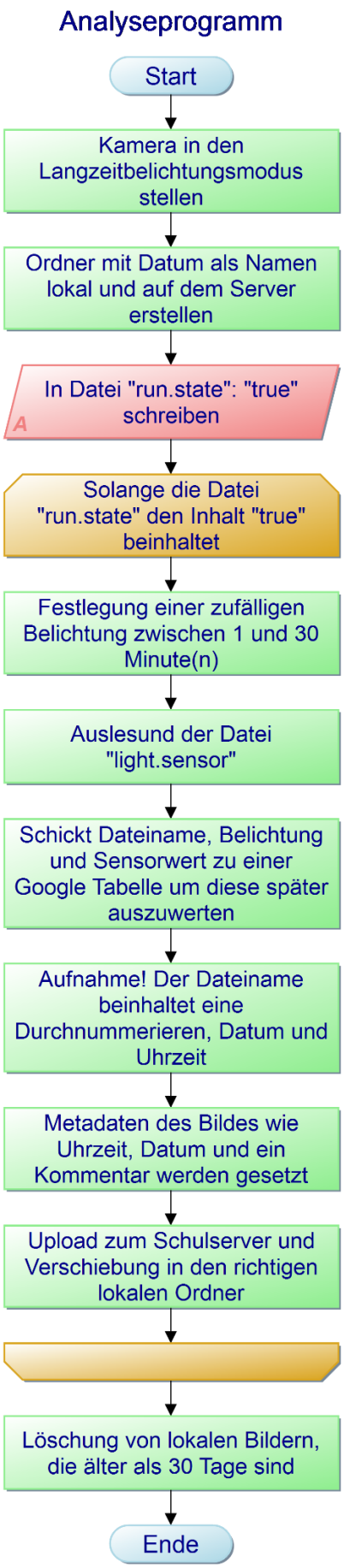

Abbildung 20 Programmablaufplan des Ermittlungsprogramms

komplexes C++ Programm zu entwickeln. Es ist zwar keine offizielle Software von Canon, jedoch arbeitet das Programm trotzdem zuverlässig.

Der Sensor lässt sich am besten mit einer Bibliothek der Firma Adafruit auslesen. Diese ist in Python3 geschrieben und erlaubt ein einfaches einstellen und auslesen.

Der Motor wird auch durch ein Python3 Programm für eine Minute beschleunigt. Dafür gibt es native Bibliotheken und der Systemdienst Cronjob erlaubt ein einmaliges Ausführen pro Tag um 15 Uhr.

Um verwechslungen zu vermeiden wird auch noch nach 0:00 das Datum für den Ordner verwendet.

Diese Datei wird zum Stoppen des Programms abgeändert

2.4.1. Sensorprogramm

Das Python3 Programm für den Sensor läuft dem Programmablaufplan der Abbildung 19 ab. Jede Sekunde wird ein Messwert in eine Liste hinzugefügt, woraus dann der Mittelwert über 5 Minuten berechnet und in eine Datei geschrieben wird.

2.4.2. Ermittlungsprogramm

Das Ziel des aktuell laufenden Programms ist eine Umgebungslicht-Belichtungszeit Kurve zu erstellen für die optimale Belichtung. Dafür wählt das Programm zufällig eine Belichtungszeit zwischen 1-30 Minute(n) und schreibt die Sensordaten sowohl in die Datei, als auch auf eine online Datenbank. Nach ein paar Tagen trägt man dann die Sensor- und Belichtungsdaten, welche zu einem gut belichtetem Bild führen, in ein Diagramm und ermittelt am Ende die gesuchte Kurve.

Der Kommentar beinhaltet die Sensordaten

Der Grund für diese Methode ist, dass das Programm nur ein paar Wochen laufen muss, um eine Kurve zu erstellen. Ein „intelligenteres“ Programm, welches automatisch die Belichtung schon analysiert wäre zu aufwendig gewesen, da wenn die Kurve einmal ermittelt ist, man keine Korrekturen mehr vornehmen muss. Gestartet und gestoppt wird das Programm durch ein Cronjob, bei dem eine Uhrzeit hinterlegt wird.

2.4.3. Dauerprogramm

Sobald die Umgebungslicht-Belichtungszeit Kurve erstellt ist, wird durch ein Cronjob jede Minute gecheckt, ob es schon dunkel genug ist, indem der 5 Minuten Mittelwert geprüft wird. Wenn es soweit ist, wird das gleiche Programm wie das Analyseprogramm ausgeführt, mit dem Unterschied, dass die Belichtungszeit berechnet wird und der Stopp durch ein Check, ob es noch dunkel genug ist, bestimmt wird.

3. Auswertung

Die Auswertung der Bilder ist komplex und im folgenden Abschnitt wird lediglich das Prinzip erklärt: Die Flugbahn einer Feuerkugel ist in der Erdatmosphäre nahe zu linear. Nun bekommt man durch den Leuchtstreifen eines Fotos eine Ebene, in der die Flugbahn der Feuerkugel liegt. Um jetzt diese ausfindig zu machen, braucht man die Schnittlinie mit noch einer weiteren Ebene einer anderen Kamera. Nun kann man den Lande Ort recht einfach durch eine dritte Ebene der Erde erlangen.

Mittlerweile weiß man, dass wenn die Feuerkugel anfängt in Teile zu zerbrechen und auch nicht mehr beobachtbar ist, eine parabelförmige Flugbahn bekommt. Das bedeutet, dass der berechnete Punkt weiter weg vom wahren Auftrittspunkt sein muss.

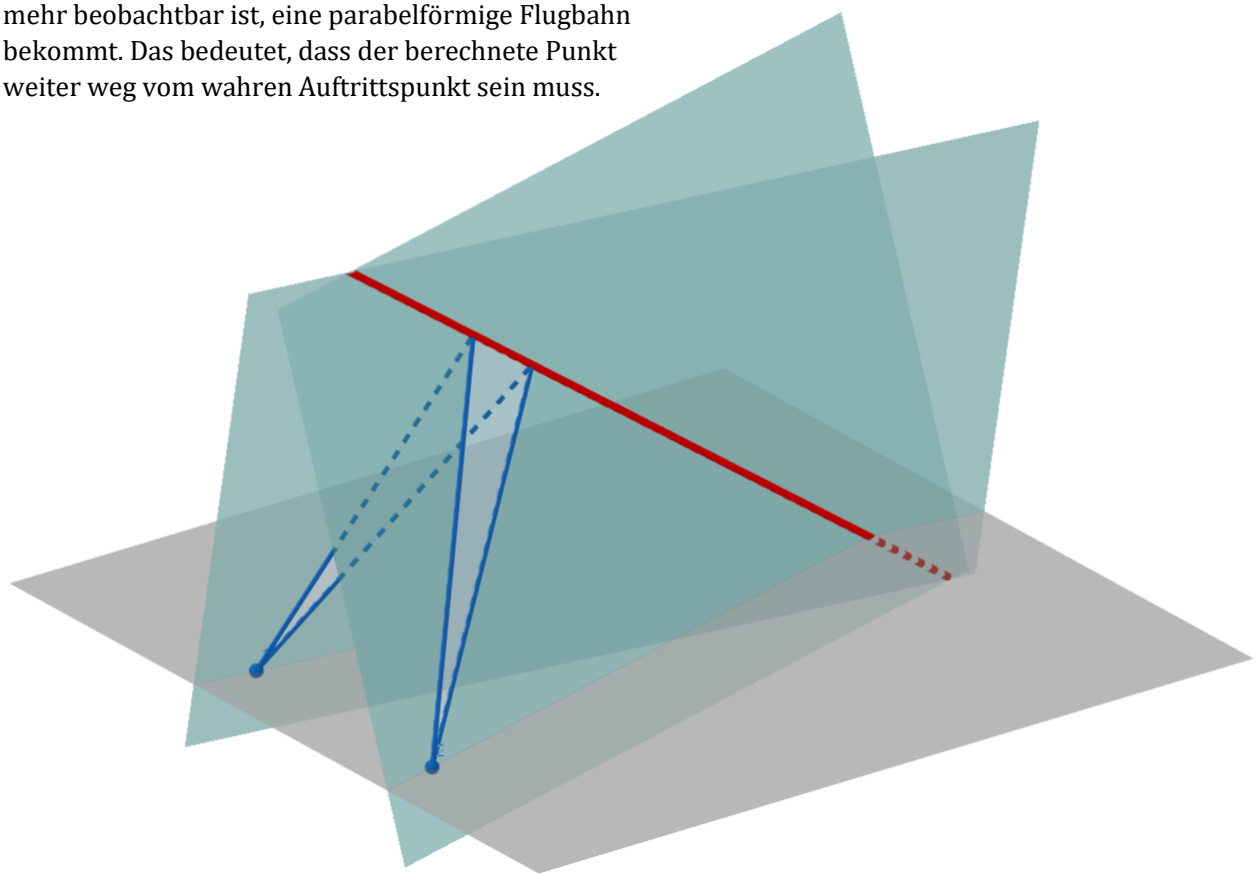

Abbildung 21 Grafik zur Veranschaulichung der Berechnung. Die rote Gerade beschreibt die Flugbahn und die Schnittgerade, die blauen Dreiecke die Beobachtung und die grünen Ebenen die berechnete Ebene um die Schnittgerade zu bestimmen. Die graue Grundebene ist die Erde.

4. Zusammenfassung

Mein Ziel einer Automatisierung der Meteorkamera habe ich fast erreicht. Zusätzlich zu dem System des Vorjahres, welches jede Nacht mehrere Bilder aufgenommen hat, werden bald die Bilder von einer deutlich besseren Qualität profitieren. Zudem sind kleinere Fehler, wie die grobe Justierung des Motors, behoben worden. Letzte Optimierungen und neue Feature werden das System in Zukunft noch bereichern.

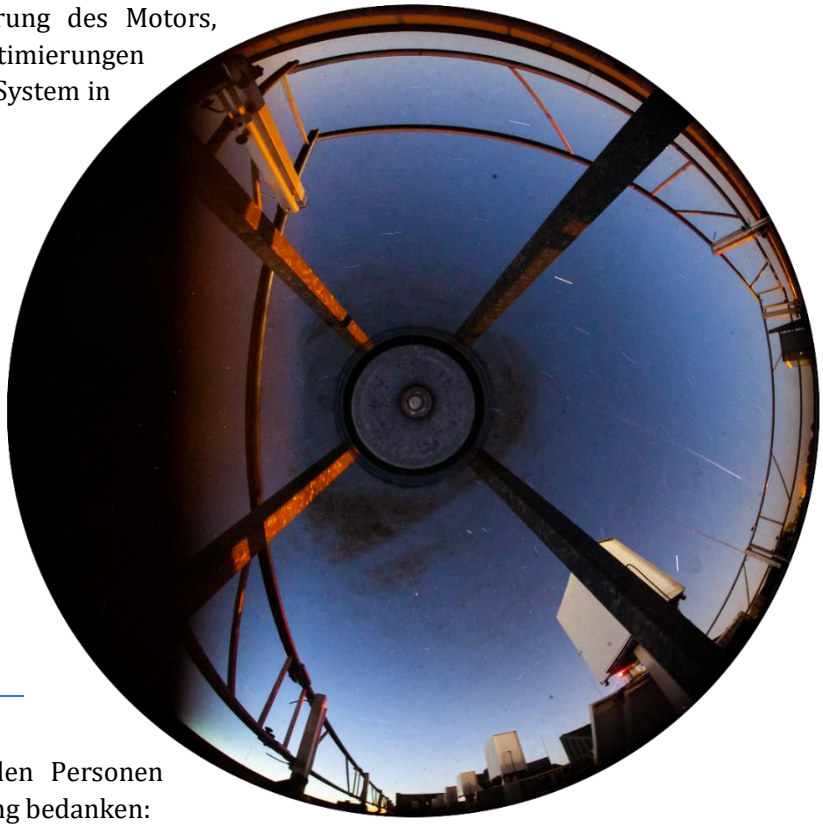

Abbildung 22 Rechts sieht man einen sehr hellen Iridium 58 Flare mit einer Helligkeit von -6.5mag. Das Bild wurde am 6.5.2018 um 20:25UT mit einer Belichtung von 15 Minuten aufgenommen. Links der Schatten ist vom stehengebliebenem Shutter.

5. Danksagung

Ich möchte mich bei folgenden Personen herzlichst für ihre Unterstützung bedanken:

Bernd Koch ist der Projektkursleiter der Astronomie und ich durfte unter anderem eine sehr interessante Ausbildung an der Sternwarte genießen. Zudem sind viele gute Ideen der Software ihm zu verdanken. Mein Physik Leistungskurslehrer ist **Michael Winkhaus** und er hat mich bei diesem Projekt nicht nur unterstützt, sondern brachte mich auch erst auf die Idee. Außerdem bedanke ich mich bei **Dieter Heinlein** für den informativen Emailaustausch und seiner Texte rund um das Meteoritenortungsnetz. Bei der Planung und beim Bau der Holz Box hat mir mein Vater **Peter Pajak** geholfen. **Florian Gerbracht** habe ich das Frequenzmessgerät zu verdanken. Zuletzt danke ich **Sophia Haude**, da sie das Meteorprojekt 2007 ins Leben gerufen hat und der ich das Gestell und den Spiegel zu verdanken habe.

Quellen

- **Meteore - Eine Einführung für Hobby-Astronomen** (S. 503f, S. 296) – Rendtel Jürgen und Arkt Rainer – Oculum-Verlga, 2012 – 1.1 Begriffserklärung und zur Fußnote 1
- **Astronomie in Theorie und Praxis** (S. 8) – Wischnewski Erik – Kaltenkirchen, 2013 – 1.1 Begriffserklärung
- **spacewaether.com** (22. April 2018) – Phillips Dr. Tony - Aufgerufen am 11. Januar 2019 www.spaceweather.com/archive.php?view=1&day=05&month=01&year=2019 – Zu 1.1 Begriffserklärung
- **NASA's All Sky Fireball Network** – Aufgerufen am 24. April 2018 - fireballs.ndc.nasa.gov – 1.1 Begriffserklärung
- **Feuerkugelnetz** – Aufgerufen am 4. Mai 2018 - www.dlr.de/pf/desktopdefault.aspx/tabid-8909/18430_read-37907/ – Zu 1.2 Das Europäische Feuerkugelnetz
- **Meteor Neuschwanstein** – Aufgerufen am 19. Mai 2018 – http://www.dlr.de/pf/desktopdefault.aspx/tabid-8955/15458_read-37909/admin-1/ – Zu 1.2 Das Europäische Feuerkugelnetz
- **Meteoritenortungsnetz: Ergebnisse 2017** – Dieter Heinlein – Zu 1.2 Das Europäische Feuerkugelnetz
- **Bedinungsanleitung** – Canon EOS 500D – 2.2 Hardware
- **Silikon-Aderleitung SiF** – Aufgerufen am 23. Mai 2018 – https://www.faberkabel.de/de/produkte/typgruppe/5801/sif/ – Zu 2.2 Hardware
- **Gphoto.org** – Aufgerufen am 21. Mai 2018 – http://www.gphoto.org/proj/ – Zu 2.4 Software
- **Facharbeit** – Sophia Haude – Zu 3 Auswertung
- **Sterne und Weltraum 8/2009, S. 78-83** – Sophia Haude – Zu 3 Auswertung
- **CalSky** – Aus Email Verkehr mit Herrn Koch – Zu Abbildung 22

Anhang 1: Lichtsensorprogramm in Python3

```
import time

import board
import busio

import adafruit_tsl2591

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tsl2591.TSL2591(i2c)
# Initialisiert die I2C-Schnittstelle und den Sensor

#sensor.gain = adafruit_tsl2591.GAIN_LOW # 1x
#sensor.gain = adafruit_tsl2591.GAIN_MED #25x Standard
#sensor.gain = adafruit_tsl2591.GAIN_HIGH # 428x
#sensor.gain = adafruit_tsl2591.GAIN_MAX # 9876x
#sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_100MS # Standard
#sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_200MS
#sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_300MS
#sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_400MS
#sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_500MS
sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_600MS
# Es lassen sich die Verstärkung und die Belichtung ändern

light = [] # Leere Messwertliste die über 5 Minuten gemittelt wird

while True:
    visible = sensor.visible # Sichtbares Licht messen
    if visible != 0: # Wenn der Wert NICHT Null ist dann
        if len(light) >= 300: # Prüfe ob die Liste schon die 5 Minuten voll
hat (60*5 = 300 Sek)
            light.pop(0) # Lösche den ältesten Wert
        light.append(visible) # Füge neuen Wert hinzu

        file = open('/home/pi/meteorkamera/python3/light.sensor', 'w')
        file.write(str(round(sum(light)/len(light))))
        file.close()
        # Berechne Durchschnitt und schreibe ihn in die Datei light.sensor
    time.sleep(1.0) # Eine Sekunde Warten
```

Anhang 2: Ermittlungsprogramm in Bash

(Google Form Link wurde anonymisiert)

```
#!/bin/bash

gphoto2 --set-config shutterspeed=bulb # Kamera in den
Langzeitbelichtungsmodus bringen

date=$(date +'%Y-%m-%d') # Datum ermitteln und in date speichern

smbclient -A smbclient.conf //astro-server-b/aktuelleprojekte --directory
Meteorkamera -c "mkdir $date" # Ordner auf dem Server mit dem aktuellen Datum
erstellen

mkdir ../img/$date # Erstelle lokalen Ordner mit dem Datum

echo true > run.state # Datei zum stoppen

num=1 # Nummerierung

while [[ $(<run.state) == "true" ]]; do # Lasse das Programm so lange laufen,
bis stop.sh ausgeführt wird
    time=$(date "+%H-%M-%S") # UTC Zeit für den Dateiname
    timeMeta=$(date "+%Y:%m:%d %H:%M:%S") # UTC Zeit für die
Metaeigentschaften der Fotodatei
    filename=$num"_"$date"_"$time"UT" # Dateiname als Variable setzen:
Nummer_Datum_ZeitUT
    let exposure=$(shuf -i 1-30 -n 1)*60  # Wähle erstmal eine zufällige auf
die Minute gerundete Belichtung zwischen 1 - 30 Minuten  für die Analyse
    light=$(<light.sensor) # Liest die Lichtintensität aus einer Datei aus,
welche durchgängig von einem Python3 Programm beschrieben wird

    curl https://docs.google.com/forms/d/***/formResponse -d ifq -d
entry.42236738=$filename -d entry.1491568817=$exposure -d
entry.976034470=$light -d submit=Submit > /dev/null
    # Schickt Dateiname, Belichtung und Sensorwert zu einer Google Tabelle um
diese später auszuwerten

    gphoto2 --set-config eosremoterelease=Immediate --wait-event=${exposure}s
--set-config eosremoterelease="Release Full" --wait-event-and-download=5s --
skip-existing --filename "$filename.%C" # Aufnahme! Setzte Dateiname aus der
Variablen + Suffix zusammen

    exiv2 -v -M"set Exif.Photo.DateTimeOriginal '$timeMeta'" $filename.*
    exiv2 -v -M"set Exif.Photo.DateTimeDigitized '$timeMeta'" $filename.*
    exiv2 -v -M"set Exif.Image.DateTime '$timeMeta'" $filename.*
    exiv2 -v -M"set Exif.Photo.UserComment '$light'" $filename.*
    # Ändere die Meta-Zeit-/ und Lichtinfos der Bildateien
```

```
    smbclient -A smbclient.conf //astro-server-b/aktuelleprojekte --directory
Meteorkamera/$date -c "prompt; mput $filename.*"
    # Lade die Fotos auf den Astroserver, unterdrücke rückfragen

    mv $filename.jpg ../img/$date/$filename.jpg
    mv $filename.cr2 ../img/$date/$filename.cr2
    # Verschiebe die Bilder zum richtigem lokalem Ordner

    let "num ++" # Erhöhe die Nummerierung um eins
done

find /home/pi/meteorkamera/img/ -mindepth 1 -mtime +7 -delete
#Löscht Bilderordner die älter als 30 Tage sind
```