

view the sky with open eyes

AllskyToGo - mobile Allsky Kamera



[PointOfStart](#)
[Meine Allsky Projekte](#)
[Allsky 1](#)
[Allsky 2](#)
[Allsky Archiv](#)
[Allsky Highlights](#)
[Webcam](#)

[Links I like](#)



AllskyToGo - mobile Allsky Kamera

Allsky Kamera 1 - Astrotonne

Allsky Kamera 2 - DomeKamera

Allsky Kamera 3 - AllskyToGo

Die Idee

Jedes mal wenn ich im Urlaub oder irgendwo anders die Möglichkeit hatte, den wundervollen Nachthimmel zu beobachten, habe ich mich geärgert, dass ich keine Kamera dabei hatte, um wenigstens einen Teil des nächtlichen Spektakels festzuhalten. Egal ob in den Alpen oder am Meer, immer fehlte die Möglichkeit den Nachthimmel fotografisch festzuhalten. Sicher ist das mit einer Digitalkamera oder einem Handy heute leicht möglich. Aber meistens sind dann die Fotos falsch belichtet oder verwackelt. So wurde bei mir die Idee einer AllskyToGo Kamera geboren. Ziel sollte sein: Kamera hinstellen, Strom dran und das Gerät arbeitet. Die Bilder sollen Live betrachtet und für die spätere Auswertung gespeichert werden.

Dieses AllskyToGo Projekt hat sich über die Zeit zu meinem Lieblingsprojekt entwickelt. Warum? - weil fast alles mit einfachsten Mitteln selbst entwickelt ist und ich immer wieder fasziniert bin, was da so alles geht. Ich werde dieses Projekt etwas ausführlicher beschreiben, damit es vielleicht ein paar Nachahmer findet. Meine Software werde ich gerne zur Verfügung stellen und ich stehe auch gerne für Fragen zur Verfügung.

Die grundlegenden Anforderungen waren schnell formuliert:

- voller automatischer Betrieb und Speichern der Aufnahmen
- automatische oder manuelle Belichtung
- mobile Stromversorgung für mindestens eine Nacht
- einfache Softwarelösung mit Weboberfläche mit unabhängigen WLAN
- uneingeschränkte Mobilität
- kostengünstige Lösung

In der Bastelkiste waren noch die folgenden Teile vorhanden:

- Raspberry Pi NoIR Camera Module 8MP v2 (Sony IMX219)
- Raspberry Pi Zero W V.1.1, 1 GHz, 512 MB Ram - SD Karte 64GB
- Gehäuse für Raspberry PiZero
- 2 Powerbanks mit 2000 und 2200 mAh & USB Y-Kabel & Micro USB Kabel
- USB Netzteil (zum laden der Powerbank oder bei Netzbetrieb)
- altes Huawei Handy Mate 7
- Amazon Fire HD8 Tablet
- 25mm UV Filter
- Kunststoffbox für Skatkarten
- Kleinmaterial wie Edelstahlschrauben etc.
- flexibles Mini Fotostativ

Der Spaß beginnt!

Raspberry Pi Zero W - Hardwarearbeiten

Zuerst wird der Raspberry mit der Kamera verbunden und in sein Gehäuse eingebaut. Das Standardgehäuse ist nicht gerade wetterfest. Dieses Problem werden wir aber später mit der Skatkartenbox lösen. Die Stromversorgung kann über die Powerbanks erfolgen. Alternativ kann in der Bastelphase aber auch das USB Netzteil verwendet werden. Da ich nur zwei kleine Powerbanks mit 2000 & 2200 mAh hatte, habe ich dieses parallel mit einem USB-Y-Kabel zusammengeschaltet. Dieses hat auf der einen Seite zwei USB-A Stecker (kommen in die Powerbanks) und auf der anderen Seite eine USB-A Buchse. Hier wird dann das normale Micro USB Kabel für die Stromversorgung des Raspi angeschlossen. Die zwei kleinen Powerbanks reichen bei mir für 10-12 Stunden, halten also locker eine Nacht durch. Der Zukauf einer großen Powerbank würde die Möglichkeiten enorm erweitern.

Das Ganze mit einer Powerbank sieht dann etwa so aus.



Einrichten des Raspberry Pi

Als Nächstes muss der Raspberry mit der nötigen Software versorgt werden. Dazu wird auf der SD Karte aus der Teilleiste Raspian OS in der 32bit Version installiert. Den Raspi Fans unter uns werden die Schritte geläufig sein, für die anderen hier ein paar Erklärungen.

- Raspian OS 32bit installieren

Das Raspian OS installiert man am einfachsten über den Raspberry Pi Imager. Hier wird festgelegt, dass SSH (für FTP oder Terminal Zugriff) aktiviert und eine WLAN Verbindung (Parameter angeben) hergestellt werden soll. Zudem ist hier der Benutzer (EigenerUserName im weiteren Text) und ein Passwort zu vergeben, da Raspian den bisherigen Standardbenutzer PI nicht mehr verwendet. Nachdem der Imager das Image geladen und auf die SD Karte geschrieben hat, kann die SD Karte in den Raspberry Pi gesteckt werden.

Tipp an dieser Stelle - nach der vollständigen Einrichtung und wenn alle Parameter der Kamera stehen, kann man sich ein Image der SD Karte ziehen (z.B. mit [Etcher](#)) um ein Backup zu haben oder die Karte zu klonen.

- DHCP IP Adresse für neues Gerät am WLAN Accespoint bestimmen

Als Nächstes wird der Raspi mit Strom versorgt. Dieser startet nun und meldet sich am Router im WLAN an. In der Routeroberfläche findet man nun die dabei vergebene IP Adresse. Bei Fritzboxen (andere Router ähnliches Vorgehen) kann man am Besten gleich einstellen, dass dieses Gerät immer die selbe IP erhält. Dadurch werden die weiteren Schritte einfacher.

- SSH Verbindung mit Putty herstellen und anmelden

Nun wird mit Putty ([Putty](#)) eine SSH Verbindung hergestellt um den Raspberry Pi an der Konsole weiter konfigurieren zu können.

- RASPI-Config

Nun wird an der Konsole `raspi-config` aufgerufen. Hier wird VNC aktiviert und die Option, das Dateisystem auf die gesamte Speicherkarte zu erweitern. Beide Optionen sind nach einem Neustart aktiv. Je nach persönlichen Empfinden kann noch aktiviert oder deaktiviert werden, was sonst noch benötigt wird.

- VNC

Wer lieber über einen Desktop arbeitet, kann jetzt VNC konfigurieren, den Anderen sei es empfohlen, da ich manchmal darauf Bezug nehme. Zunächst wird der kostenlose [RealVNC Viewer](#) heruntergeladen und auf dem PC, mit dem man auf den Raspberry zugreifen will, installiert. Dann wird mit der oben ermittelten IP eine VNC Desktop Verbindung hergestellt und der Desktop des Raspberry Pi in den Systemeinstellungen auf Deutsch umgestellt. Bei Bedarf kann die VNC Resolution über `raspi-config` am Terminal eingestellt werden.

- Webserver

Die AllskyToGo Kamera benötigt für die Software einen Webserver. Die Installation auf dem Raspi ist im Terminal schnell erledigt mit den folgenden Befehlen:

```
sudo apt update && sudo apt upgrade -y
sudo apt install apache2 -y
```

Test ob Webserver funktioniert: Im Browser aufrufen `http://IP-der-AllskyTOGo-Kamera/phpinfo.php` aufrufen. Es werden PHP Informationen angezeigt.

- Noch ein paar Systemeinstellungen

Prüfen ob der eigene User in der Gruppe www-data ist

Befehl: `id EigenerUserName`

sonst hinzufügen mit: `sudo usermod -a -G www-data EigenerUserName`

anlegen von zwei Verzeichnissen

```
mkdir /home/EigenerUserName/allskytogo
mkdir /home/EigenerUserName/allskytogo/pictures
```

Zugriffsrechte festlegen

```
sudo chmod -R 777 /home/EigenerUserName/allskytogo
sudo chmod -R 777 /var/www/html/
```

Achtung! Ich weise darauf hin, dass die beiden letzten Befehle die Zugriffsrechte komplett für jedermann öffnen. Das sollte man nicht tun, wenn der

Raspberry Pi mit dem Internet oder irgendwelchen produktiven Systemen verbunden ist. Warum mache ich das? Ich will es mir einfach machen und Fehlermeldungen oder Programmabbrüche aufgrund von fehlenden Zugriffsrechten vermeiden.

- Software kopieren

Per FTP (ich nutze Filezilla) den Inhalt des ZIP Archivs mit der Software (wird später auf dieser Seite bereitgestellt) in das Verzeichnis //var/www/html/ auf dem Raspberry kopieren. Die FTP Verbindung kann als SFTP - SSH Verbindung mit der IP des Raspberry und den angelegten Userdaten hergestellt werden.

- Cronjob für die Aufnahmesoftware

Um die Aufnahmen automatisch zu starten, wird ein Eintrag in der Crontab des Users benötigt. Dazu gibt man im Terminal den folgenden Befehle ein:

```
crontab -e
```

Am Ende der dann erscheinenden Datei fügt man dann die folgenden Zeilen hinzu:

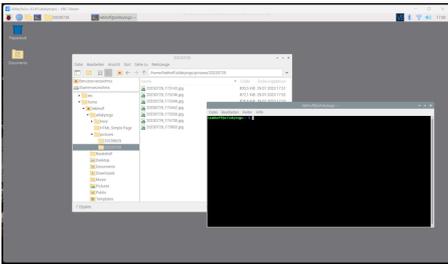
```
#Bei reboot wird der AllskyToGo Kamera Job gestartet
@reboot /usr/bin/php /var/www/html/allskyrun.php EigenerUserName
```

Zur Kontrolle des Cronjobs im Protokoll kann man im Terminal eingeben:

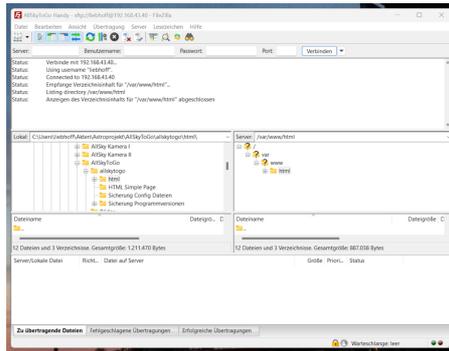
```
grep CRON /var/log/syslog
```

Zwischenstand

Was haben wir nun erreicht? Wir haben einen funktionsfähigen Raspberry Pi mit einer Softwareinstallation, die automatische Aufnahmen ermöglicht. Zur Software schreibe ich später mehr und sehr ausführlich.



Remote Zugriff via VNC



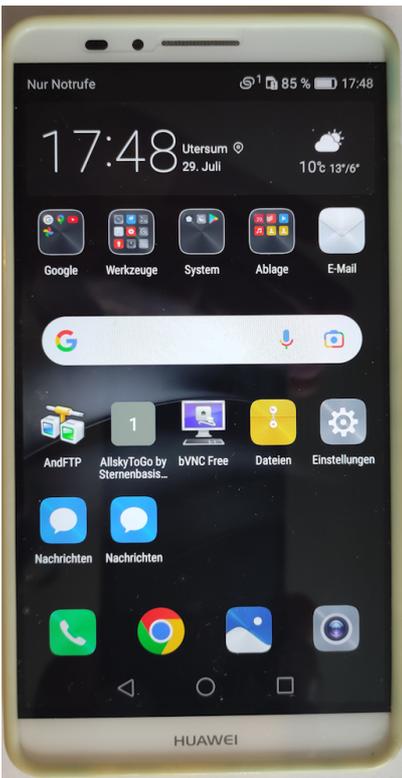
Dateizugriff via FTP



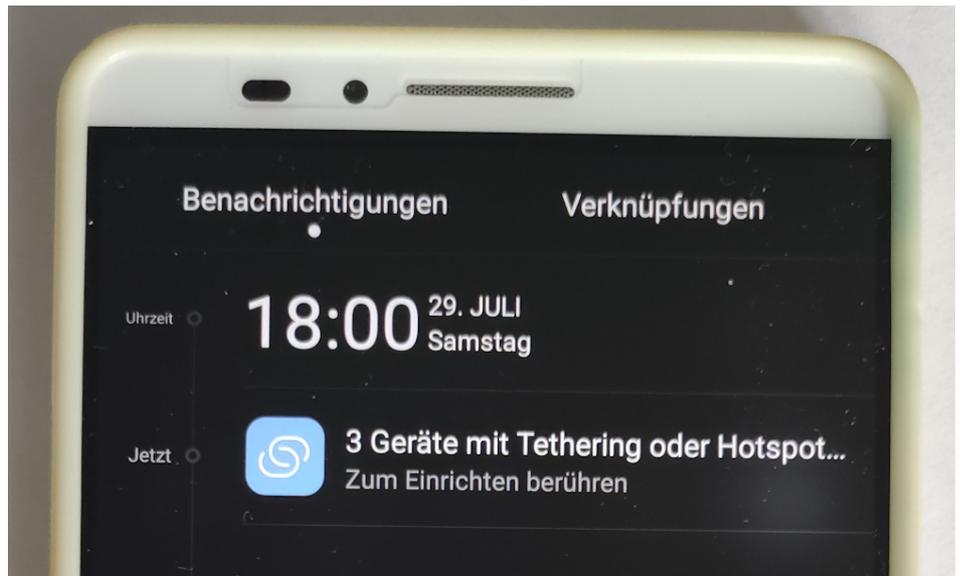
Eine Weboberfläche für die Aufnahmesteuerung und den Live View

Was mir jetzt noch fehlte, war die Mobilität und letztendlich noch die finale Hardwarelösung mit Gehäuse, Stromversorgung und Stativ.

Um mobil zu werden, sollte die AllskyToGo Kamera nicht mehr im heimischen WLAN eindocken, sondern auf einem ausgedienten Handy, welches als mobiler Hotspot konfiguriert wurde. Eine SIM Karte in dem Gerät ist dafür nicht erforderlich und ohne SIM Karte ist auch kein unerwünschter Internet Zugang möglich. Auf dem Handy wurden dann gleich noch ein VNC Client (bVNC Free) und ein FTP Client (AndFTP) installiert. Beide Softwareversionen funktionieren auch mit älteren Android Versionen. Diese sollen zusätzlich einen Zugriff auf den Kamera Raspi ermöglichen. Dazu wurde ein Link auf die Startseite der Websoftware gelegt, so dass auch darüber die Konfiguration und Steuerung der Kamera möglich ist.



Oberfläche des Hotspot Handys



3 Geräte (Kamera, Tablet und PC) am Hotspot eingedockt

Die AllskyToGo Kamera war/ist nun bereits mit zwei WLAN Netzwerken vertraut und hat die Zugangsparameter gespeichert. Das kann auch so bleiben und es können weitere WLAN Netzwerke hinzu kommen, wenn z.B: der Standort gewechselt wurde oder man sich z.B. mit einem Hotel WLAN verbinden will. Neue WLAN Netzwerke verbindet man, indem man zuerst eine Verbindung über den Handy Hotspot herstellt und dann über die VNC Oberfläche mit dem neuen WLAN verbindet. Der Wechsel zwischen den verschiedenen WLAN Netzwerken erfolgt dann meist automatisch, wenn man in den Bereich eines bekannten Netzwerkes kommt. Eventuell ändern sich dadurch IP Adressen, so dass man die VNC oder Weboberfläche entsprechend aufrufen muss.

Und da im Alter der Bildschirm eines Handys nie groß genug ist, habe ich mich entschieden, ein altes Amazon Fire HD8 Tablet als Endgerät umzurüsten. Da aber FireOS nicht ohne Umwege jede Android App unterstützt, war die erste Aufgabe dem Amazon Gerät ein wenig die Google Welt nahe zu bringen. FireOS ist auch nichts anderes als Android mit spezieller Oberfläche, sollte also gehen.

Wie das geht steht sehr schön erklärt hier: [Play Store auf Fire-Tablet installieren: Schritt für Schritt erklärt](#)

Auch dieses YouTube Video von Cubi Reviews kann eine Hilfe bei der Umrüstung sein: ["echtes" Android auf dem Amazon Fire HD8 | Play Store | Nova Launcher | Tutorial.](#)



Amazon Fire HD8 mit Google Funktionen



Die Weboberfläche der AllskyTogo Kamera auf dem Fire HD8

Hardware die Zweite!

Das Standardgehäuse des Raspberry Pi Zero W ist nicht gerade "Allwettertauglich". Ich habe mich daher entschieden, mit Bordmitteln das Ganze etwas aufzupeppen. Da ich noch einen 25mm UV Filter von Opteka hatte und dieser gut als Objektivschutz geeignet ist, habe ich eine alte Kunststoff Skatkartenbox genommen und ein Loch hineingeschnitten. In dieses wurde dann der UV Filter geklebt. Da Ganze muss natürlich so angebracht werden, dass die Linse der Raspberry Kamera hinter dem UV Filter zu liegen kommt. Damit das Raspi Gehäuse nicht in der Box verrutscht, wurde es in der Box mit ein paar Schaumstoffstücken fixiert. Diese sind mit doppelseitigem Klebeband in der Box fixiert.

An den Seiten der Boxteile wurde dann je ein Schlitz geschnitten, so dass man nach dem Zusammenfügen der beiden Hälften einen Kabeldurchlass hat. Vier Löcher an den Ecken dienen dazu, die Hälften am Ende mit Edelstahlschrauben zusammenzuhalten.

Die beiden Powerbanks werden bisher einfach in eine Plastiktüte gesteckt und mit einem Gummi zusammengehalten und verschlossen (mehr oder minder). Es führt nur das Kabel zur Kamera. Das ist ein für mich erstmal hinreichender Wetterschutz, da der Hobbyastronom ja meistens bei Schönwetter unterwegs ist.

Klar hätte man ein Gehäuse aus dem 3D Drucker ziehen können, aber erstens habe ich keinen 3D Drucker, da sich mir die Notwendigkeit noch nicht wirklich erklärt hat und zweitens finde ich die Verwendung von irgendwelchen "Altteilen" sehr viel nachhaltiger.

Da das Gehäuse nicht irgendwo rum liegen soll, habe ich noch ein altes Ministativ mit flexiblen Beinen im Projekt aufgenommen. Die Kamerahalterung wurde dabei ebenfalls fest am Kameragehäuse angeschraubt, so dass ich nun je nach Bedarf die Kamera am Stativ befestigen und dieses dank der flexiblen Beine überall aufstellen oder auch an einem Ast oder Balkongeländer anbringen kann. Wer sich bei den vorangegangenen Fotos bisher gewundert hat, warum auf der Kamera ein Aufkleber mit der Aufschrift "oben" prangt, dem sei erklärt, dass sich hier die Oberkante des von der Kamera aufgenommenen Bildes befindet. Wenn man die Kamera beim Aufstellen mit dem Schriftzug "oben" nach Norden ausrichtet, erhält man Bilder mit geografischer Orientierung.

Eine kurze Fotostrecke zu der Basterei ...





Die Software by Sternenbasis 39

Bis zu diesem Punkt habe ich im Wesentlichen Dinge getan und erklärt, die man so oder ähnlich schon woanders gesehen hat. Die im Folgenden beschriebene Software verwendet natürlich auch bekannte Elemente, z.B. um die Kamera und Aufnahmen zu steuern. Die Grundidee der AllskyToGo Kamera, wie ich sie mir vorgestellt habe, und die Anforderungen die ich daran habe, haben aber dazu geführt, dass ich eine Software geschrieben habe, die meinen Vorstellungen und Zielen entspricht und somit ein persönliches Produkt darstellt.

Meine Anforderungen an die Software kann man wie folgt zusammenfassen:

- voller automatischer Betrieb und Speichern der Aufnahmen
- einfache Softwarelösung mit Weboberfläche zur Steuerung der Aufnahmen
- automatische oder manuelle Belichtung in Abhängigkeit von Tag und Nacht und den verschiedenen Dämmerungsphasen
- variable Standortdaten für die Ermittlung der Dämmerungsphasen
- Live View auf das letzte aufgenommene Bild
- Konfiguration von verschiedenen Aufnahmeparametern (ISO/GAIN, Belichtungszeit, Aufnahmefrequenz (Abstand zwischen zwei Aufnahmen), Weißabgleich)

Funktionsprinzip

Für die Steuerung der Raspberry Kamera wird LIBCAMERA aus dem Raspberry Pi Projekt verwendet. ([Introducing the Raspberry Pi Cameras](#)) Bei Start der AllskyToGo Kamera wird ein Cronjob gestartet, der die bestehenden Konfigurationsdateien liest, die astronomischen Dämmerungsphasen ermittelt und mit den hinterlegten Aufnahmeparametern ein Foto aufnimmt. Dieses Foto wird im Ordner `/home/EigenerUserName/allskytogo/pictures` abgelegt. Gleichzeitig wird es in das Webserververzeichnis kopiert, so dass daraus der Live View generiert werden kann. Die gespeicherten Fotos bekommen den üblichen Datum/Zeitstempel als Namen. Dieser Job läuft in einer Endlosschleife und nimmt in der hinterlegten Aufnahmefrequenz immer wieder Bilder mit den jeweils aktuellen Parametern auf. (in Abhängigkeit von den Dämmerungsphasen) Der Job kann unterbrochen werden, für einen Neustart muss aber die ganze AllskyToGo Einheit neu gestartet werden. Gleichzeitig startet auf dem Raspberry der Webserver, der den Live View präsentiert, sowie die Verwaltung der Parameter ermöglicht. Die meisten dort vorgenommenen Änderungen werden sofort für das nächste Foto berücksichtigt.

Die Oberfläche der AllskyToGo Kamera sieht wie folgt aus - Die Erklärung der einzelnen Menüpunkte folgt im Anschluss, für die vollständige Anzeige der Seitenvorschau bitte auf das Bild klicken.

AllskyToGo by Sternenbasis 39

my Sky above

Startseite
Konfiguration
Simulation
Datum und Uhrzeit
System
Allsky LOG
Sat Jul 29 17:56:09 CEST 2023

AllSkyToGo

Sie sehen das Bild image.jpg (letzte Aufnahme) - automatische Aktualisierung alle 10 Sekunden -
Oben ist Norden. Westen ist rechts.

Kamera: RaspberryPi CAM.V2 NoIR 8MP Sony IMX219 Objektiv: f 3.04 mm 1:2.0
 Brennweite: 3.04 mm Belichtung: 0 Sekunden (18222/1000000) Blende: 1:2.0 ISO: 200
 Aufnahmezeitpunkt: 29.07.2023 17:55:58



Startseite

Die Startseite zeigt im oberen Bereich Parameter zur Kamera und der letzten Aufnahme an.

Danach folgt das letzte aufgenommene Bild, wie es auf den Webserver kopiert wurde.

Aus der letzten Aufnahme werden die EXIF Daten ausgelesen und unter dem Bild ausgegeben.



Konfiguration

Es gibt vier verschiedene Konfigurationsdateien. Diese sind im oberen Bereich der Seite mit einer Beschreibung aufgelistet. Daneben ist ein Symbol (Buch). Wenn dieses angeklickt wird, kann die jeweilige Datei in einem Editorfenster bearbeitet werden. Bei der Syntax handelt es sich um PHP. Im Wesentlichen werden Variablen definiert und gesetzt, die dann vom Programm verarbeitet werden.

Mit dem Speichern der Dateien werden die Einstellungen für den nächsten Lauf des Cronjobs sofort wirksam. Es erfolgt keine Prüfung auf Sinnhaftigkeit oder Richtigkeit der Eingaben. Man muss wissen, was man tut!

Die Konfigurationsdateien werden weiter unten noch genauer erläutert.



Simulation

Diese Seite dient zur Überprüfung der Ergebnisse aus der eingestellten Konfiguration. Es wird keine Aufnahme durchgeführt, sondern nur die ermittelten Aufnahmeparameter angezeigt.

Als erstes wird das aktuell verwendete Datum und die Zeit angezeigt und die daraus ermittelte Dämmerungsperiode. Es wird unterschieden zwischen: daytime, night, civil, nautic, und astro. Zusätzlich werden noch einmal die genauen Daten der astronomischen Dämmerungsperioden für den ermittelten Standort angezeigt. Wenn eine Dämmerungsperiode mit "1970.01.01 01:00:01" angezeigt wird, heißt das, dass diese Dämmerungsphase im Moment am aktuellen Standort nicht erreicht wird. Z.B. Wird hier im Norden in den "weißen Nächten" die astronomische Dämmerung nicht erreicht, weil die Sonne nicht weit genug unter den Horizont sinkt. Über das Eingabefeld im Kopf, kann die Simulation für ein anderes Datum vorgenommen werden. Dieses verstellt nicht die Uhr des Raspi.

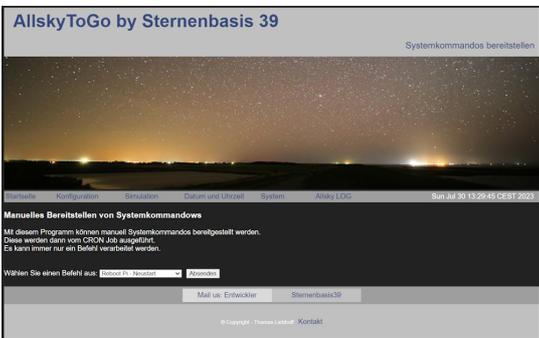
Als nächstes folgen die ermittelten Werte für die Aufnahme. Diese würden im Fall einer zum Zeitpunkt stattfindenden Aufnahme benutzt. Die erzeugten Shell Befehle sind die selben Befehle, die auch der Cronjob ausführen würde.



Datum und Uhrzeit

Auf dieser Seite können Datum und Uhrzeit für den Raspberry Pi eingestellt werden. Die Werte sind eventuell veraltet, da der Pi Zero keine interne Uhr besitzt und eventuell keine Internet (NTP) Verbindung besteht. Wenn die Uhrzeit nicht über einen NTP Server oder das Internet aktualisiert werden kann, wird der letzte bekannte Wert verwendet. Dieses ist z.B. der Fall, wenn ich das Handy als Hotspot verwende. Dieses hat keine SIM Karte und somit kein Internet.

Ist die Datums- oder Zeitangabe falsch, erhalten Dateien falsche Namen. Die Berechnung der astronomischen Dämmerungsdaten funktioniert zwar, aber es kommen natürlich falsche Ergebnisse heraus, die dann zur Verwendung von falschen Belichtungsparametern führen.



System

Unter diesem Punkt werden einige Systembefehle bereitgestellt, die eventuell für die Arbeit mit der AllskyToGo Kamera erforderlich sind. Diese werden vom Cronjob verarbeitet. Wenn der Parameter Display="ON" ist, erfolgt eine eventuelle Ausgabe in das Systemprotokoll.



Allsky LOG

Hier wird das Protokoll des Cronjobs angezeigt. Normalerweise enthält dieses nur Informationen zu den aufgenommenen Fotos. Wenn der Parameter Display="ON" ist, werden alle Ausgaben des Cronjobs protokolliert. Das kann hilfreich sein, wenn Fehler gesucht werden oder man einen genaueren Blick auf einzelne erzeugte Parameter haben möchte. Display="ON" kann in der Konfiguration gesetzt werden, Standard ist OFF.

Konfigurationsdateien im Detail

In den Konfigurationsdateien sind alle Zeilen mit "//" nur Kommentare.
 Alle Bezeichner, die mit einem "\$" beginnen sind Variablen, deren Namen nicht verändert werden dürfen.
 Zusätzliche eigene Kommentare sind erlaubt.
 Eigene Variablen auch, diese werden aber logischerweise nicht im Programm verwendet, solange der Programmierer (also ich) das nicht vorsieht.

allskytogo.config - Programmsteuerung der AllSkyToGo Kamera

Die meisten der Parameter erklären sich durch ihren Namen oder den Kommentar.
 Einige Parameter haben nach dem eigentlichen Namen z.B. \$myGain noch einen Zusatz z.B. "Astro", dieses stellt dann immer den Wert dar, denn die Variable in der jeweiligen Dämmerungsperiode annehmen soll. Variablennamen ohne Zusatz stellen die Werte für die Nacht dar.

Die Variable \$myAutoMode legt fest ob manuelle Werte je Dämmerungsperiode benutzt werden sollen oder ob im Fall von \$myAutoMode="ON" die automatische Belichtung der Raspberry Kamera benutzt werden soll.

Der Parameter \$Display="OFF" steuert nur die Ausgabe des Cronjobs in die Log-Datei. Die Anzeige am Bildschirm erfolgt immer.

Bei den Kameraparametern sollte darauf geachtet werden, dass nur Werte eingetragen werden, die von der Kamera auch unterstützt werden. Hierzu sollte man die Daten der verwendeten Kamera recherchieren.

```
// -----
// Benutzer
// -----

$Benutzer="EigenerUserName";

// -----
// Anzeige Ausgaben
// -----

$Display="OFF";

// -----
// Kameraname
// -----

$myKamera="RaspberryPi CAM.V2 NoIR 8MP Sony IMX219";

// -----
// Verarbeitungsparameter
// -----

// AUTO Mode - im Auto Mode werden die Werte für die Parameter myGain, myShutter, myMetering nicht berücksichtigt
// Belichtung wird automatisch von der Kamera gewählt. AWB (WeiBabgleich) wird berücksichtigt.
// Aufnahmeparameter (Parameter ohne Zusatz entspricht Wert für Nacht)

$myAutoMode="OFF";
$myAutoModeAstro="OFF";
$myAutoModeNautic="OFF";
$myAutoModeCivil="ON";
$myAutoModeDaytime="ON";

// Datei auf dem internen Webserver ablegen
$copyToWebServer="YES";
$WebServerPfad="/var/www/html/";
$WebServerFileName="image.jpg";

// auch die Datei unter dem original Namen kopieren
$WebServerFileNameOriginal="NO";

// -----
// Bildparameter
// -----

$myWidth="3280";
$myHeight="2464";
$myOutputfile="allskytogoout.jpg";

// -----
// Aufnahmeparameter (Parameter ohne Zusatz entspricht Wert für Nacht)
// -----
```

```
// ISO ergibt sich aus GAIN ISO=GAIN*100
$myGain="6";
$myGainAstro="6";
$myGainNautic="6";
$myGainCivil="1";
$myGainDaytime="1";

// Belichtungszeit in Microseconds 1/60 = 0,0166666 1/30 0,0333333 1/125=0,008 1/250=0,004 1/500=0,002
// Maximum exposure times (seconds) für Pi Kamera V2 11.76 Sekunden
$myShutter="10000000";
$myShutterAstro="10000000";
$myShutterNautic="10000000";
$myShutterCivil="8000";
$myShutterDaytime="8000";

// Manueller Weißabgleich (rot, blau)
// gain must be between 0.0 and 8.0. Typical values for the gains are between 0.9 and 1.9 (0,0 wie auto)
$myAwbgains="0,0";
$myAwbgainsAstro="0,0";
$myAwbgainsNautic="1.0,1.2";
$myAwbgainsCivil="1.0,1.2";
$myAwbgainsDaytime="1.0,1.2";

// Wartezeit zwischen zwei Aufnahmen in Sekunden
$wartezeit="2";
$wartezeitAstro="2";
$wartezeitNautic="5";
$wartezeitCivil="30";
$wartezeitDaytime="60";

// Set the metering mode of the AEC/AGC algorithm, the available parameters are:
// centre - Center metering (default)
// spot - spot metering
// average - average or full frame metering
$myMetering="centre";

// Ende Konfiguration
```

location.config - aktuelle Position der AllSkyToGo Kamera

In dieser Datei werden die Standortdaten eingetragen, die aktuell genutzt werden sollen. Alte Standorte können in der Datei verbleiben, falls man sie erneut benutzen möchte. Es sollten immer alle nicht benutzten Standorte mit "/" auskommentiert werden. Vergisst man einen Standort und hat somit mehr als einen Standort aktiv, werden die Werte des letzten gefundenen Datensatzes genommen.

```
// Position der AllskyToGo Kamera - wird für die Berechnung der astronomischen Daten benutzt.
// Wenn die Angaben nicht passen, kommt es zu Abweichungen, dass Programm ansich funktioniert aber trotzdem

// nicht benötigte Werte auskommentieren

// Werte für Utersum
$aktuellerStandort="Utersum";
$breitengrad=54.7111;
$laengengrad=8.4069;

// Werte für Lüsen - Hotel My Herol
// $aktuellerStandort="Lüsen";
// $breitengrad=46.7601;
// $laengengrad=11.7630;
```

cancel.config - Abbruchsteuerung

Der Kommentar und der Code sind selbsterläuternd.

```
// wenn $nochmal="NO"; wird eine mögliche Schleife im Programm allskyrun.php abgebrochen
// rücksetzen nicht vergessen YES

$nochmal="YES";
```

extra.config - extra Befehl

Der Kommentar und der Code sind selbsterläuternd.

```
// wenn die Variable EXTRABEFEHL nicht leer (Inhalt ist -leer-) ist, wird der Befehl im CRONJOB ausgeführt und danach gelöscht

$extrabefehl="leer";
```

Ich habe fertig

... Nein eigentlich nicht, man ist ja nie fertig mit seinen Bastelprojekten. Es gibt schon eine Reihe neuer Ideen (z.B. FTP Einbindung).

Ich habe für mein Projekt mit einem "kleinen" Raspberry Pi Zero W gearbeitet. Dieser hat sich als ausreichend leistungsfähig für das AllskyToGo Projekt erwiesen. Sowohl die Aufnahmen als auch der Webserver laufen problemlos. Das ganze System ist nicht sehr stromhungrig und kann auch ohne WLAN oder Handyhotspot betrieben werden. AllskyToGo Kamera mit Strom versehen und das kleine Wunder sammelt Bilder, nicht mehr und nicht weniger.

Bei der Kamera selber habe ich eine original NOIR Raspberry Kamera verwendet. Prinzipiell sollten alle Raspberry Pi Kameramodule, die durch LIBCAMERA unterstützt

werden, funktionieren. Ob mit IR Sperrfilter oder ohne (NOIR) ist Geschmackssache. Andere Kameramodule oder eine USB Kamera wurden von mir noch nicht getestet, ich kann daher dazu bisher keine Auskunft geben.

Prinzipiell sind sicher auch andere Raspberry Pi Modelle verwendbar (sofern lieferbar) jedoch steigt mit der Größe auch der Strom- und Platzbedarf. Im Interesse der kleinen portablen Reiselösung würde ich also immer eher zu den auch von mir verwendeten Komponenten tendieren. Dabei ist darauf zu achten, dass der Raspberry Pi auch ein WLAN Modul hat. Dadurch wird einiges einfacher.

Fotostrecke final

Klicken für Vergrößerung



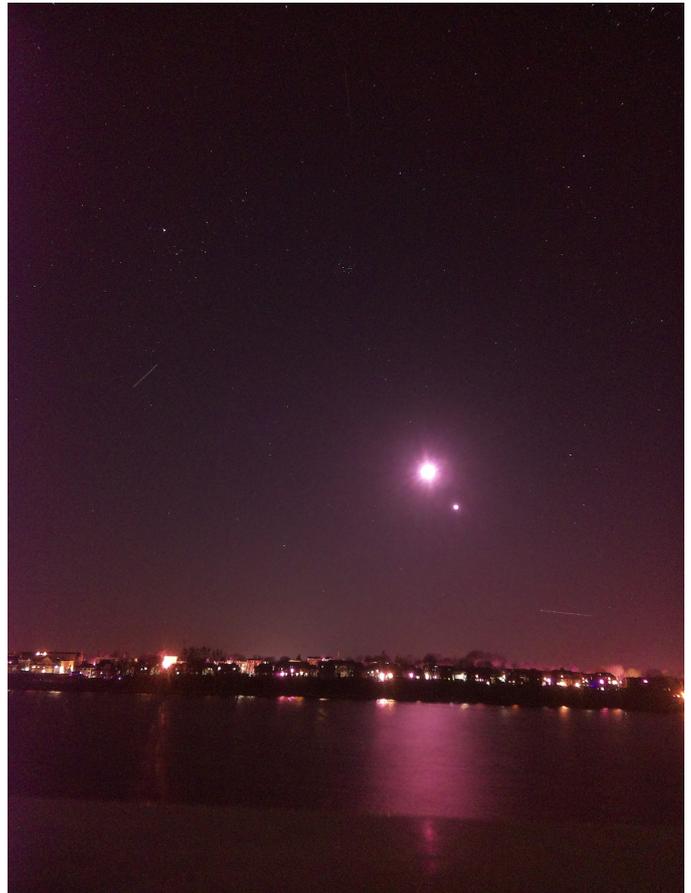
Tageslichtaufnahme - Grünlich wegen Weißabgleich



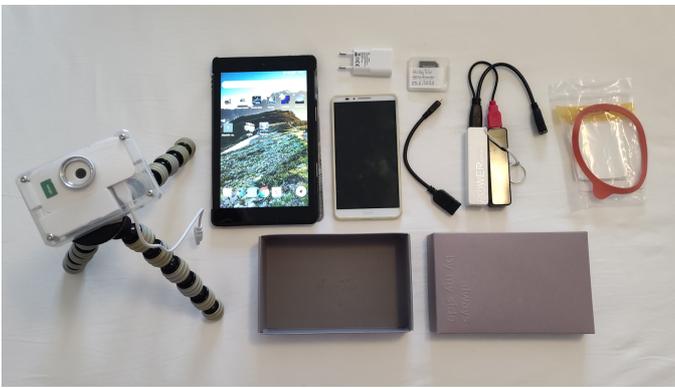
Nachthimmel bei starker Lichtverschmutzung mitten in Schwerin



Kamera auf Stativ mit Akku Pack



See und Nachthimmel bei starker Lichtverschmutzung mitten in Schwerin



Das komplette Reiseset mit Stativ



Meine alte Pappschachtel hatte zufällig den richtigen Aufdruck



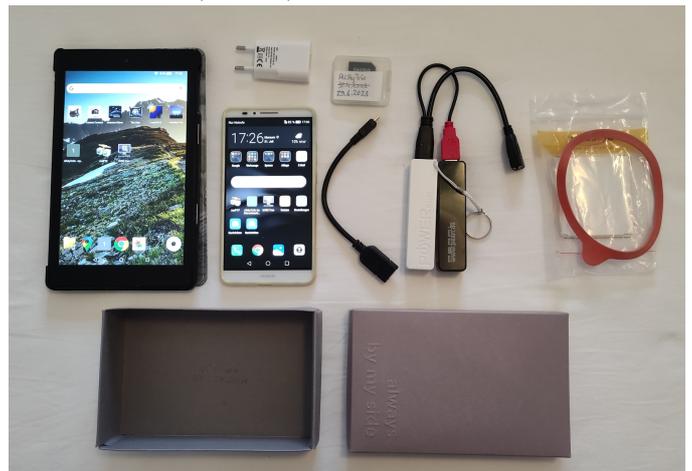
Reiseset mit Kamerabox, Stativ und Tablett



Inhalt der Kamerabox (18x12 cm), enthält alles außer Tablett und Stativ



Kamera ohne Stativ mit Akku Pack



Inhalt der Kamerabox plus Tablett

Downloads

Wer beim Lesen und Bilder schauen bis hier durchgehalten hat, darf sich hier gerne bei den zwei Downloads bedienen:

[Download PDF dieser Seite](#)

[Download der Software by Sternenbasis 39 als ZIP](#)

[Mail us: Moderator liebhoff.de](mailto:liebhoff.de)

© Copyright - Thomas Liebhoff - photographer - all pictures taken by me [Contact](#) [Legal Info](#) - [Impressum](#) [Datenschutzerklärung](#)