## Institut für technische UFO-Forschung Independent research www.ufoforschung.de

## Beschreibung zum Raspberry-System \_\_ITUF\_\_

Seit etwa 2018 bis Anfang 2020 arbeitete ich mit einem Rumpfsystem welches von Wolfgang S. (Mufon) für den Raspberry Zero entwickelt bzw. programmiert wurde. In dieser Zeit habe ich mich in das System eingearbeitet und eine verbesserte Neuentwicklung gestartet. Zusätzlich zu den Änderungen und Neuerungen wurden Teile des Python - Programm von Wolfgang S. übernommen.

Das Python – Programm wurde mit Thread's auf eine modulare Programmstruktur geändert. Nun arbeiten drei unabhängige Programme parallel auf dem Raspberry Pi.

- 1- Videoauswertung und Speicherung
- 2- Meßwertauswertung und Speicherung
- 3- Ansteuerung von externen Zusatzgeräten wie zB. hochwertige Kameras

Das hat den unschätzbaren Vorteil, das nicht ein unübersichtliches Monsterprogramm entstanden ist, sondern unabhängige Einzelprogramme, welche leicht zu ändern/erweitern sind. In diesem Zusammenhang habe ich noch eine für den Raspberry PI kompatible Relaiskarte entwickelt. Das Gesamtkonzept ist für die UFO-Forschung ein richtungsweisendes Tool.

## Das System ist als Open Source Projekt \_\_ITUF\_\_ angelegt.

## Eine kurze Programmbeschreibung:

Das Programm startet beim Einschalten des Raspberry Pi.

Belichtungsabhängig (Tag/Nacht) wird die Belichtungszeit der Einzelbilder und die Aufzeichnungsgeschwindigkeit automatisch durch das **Video-Programm** gesteuert. Bei Tag entsteht ein .H264 Video mit einem Einzelbildabstand von 2 Sekunden. Bei Nacht entsteht ein .H264 Video mit einem Einzelbildabstand von 6 Sekunden und einer Einzelbildbelichtungszeit von 6 Sekunden. Mit dieser langen Belichtungszeit werden sehr gute Nachtsichteigenschaften erzielt. Daher ist zwischen den Einzelbildern nur eine kurze Austastlücke die bedeutungslos ist.

Die lange Belichtungszeit hat einen sehr großen Nachteil, Bewegungen werden in der Bewegungsrichtung verzerrt. Das ist jedoch mit einem so kostengünstigen Gerät derzeit nicht anders zu machen. Das Video wird in der Regel auch nur als schnell zu prüfendes Übersichtsvideo benutzt und ist daher eine unverzichtbare Hilfe für die zeitnahe Prüfung der Videos. Es ist daher besonders sinnvoll den Raspberry PI im Verbund mit weiteren Kameras zu benutzen. Im Einzelfall (Prüfen ob ein Gebiet interessant ist) ist ein kleines Raspberry Pi System ohne zusätzlich Kameras durchaus das Mittel der Wahl.

Das Videoprogramm wird durch das Programm Meßwerte ergänzt.

Am Mark erhältliche Sensorplatinen für den Raspberry Pi (Das Programm ist derzeit für den ENVIR PHAT ausgelegt) zB. ENVIR PHAT oder LSM303D oder andere Produkte stellen die benötigten Meßwerte von Magnetfeld und "Gravitation" zur Verfügung. Die wichtigsten Meßwerte werden in das aktuelle Videobild eingefügt. Zusätzlich werden alle Meßwerte als TxT in eine Datei geschrieben. Diese Daten können zB. mit einem Exelprogramm visualisiert werden. Erfaßt und verarbeitet werden beim ENVIR PHAT die folgenden Daten.

Datum/Zeit, Magnetfelder X,Y,Z, Gravitation X,Y,Z (indirekte Messung durch Beschleunigungssensor), Temperatur (gibt die Wärme im Gehäuse wieder), Luftdruck. Die Geschwindigkeit der Datenaufzeichnung ändert sich mit den Erfordernissen. Im Normalbetrieb werden nur alle 2 Minuten Daten aufgezeichnet. Im Alarmfall 1 werden ca. 10 Messungen in der Sekunde aufgezeichnet. Im Alarmfall 2 werden ca. 10 Messungen in der Sekunde aufgezeichnet.

- Alarmfall 1: Wird der aktuelle Wert des Magnetfelds um einen einstellbaren % Wert überoder unterschritten (aktuell 5%), erfolgt für 5 Minuten die schnelle Datenaufnahme.
- Alarmfall 2: Ist am Ende von Alarmfall 1 das Magnetfeld nicht wieder auf dem Niveau des anfänglichen Normalbetriebs, so verlängert sich die schnelle Datenaufnahme um weiter 5 Minuten. Nach Ablauf dieser Zeit wird der dann aktuelle Stand des Magnetfeld als der Normalfall behandelt und der Modus wechselt wieder in Normalbetrieb.

Wenn das System mit zusätzlichen Kameras (zB. Sony Alpha 7s, Sony HX60V ......) erweitert werden soll, dann ist das **Programm Kamerasteuerung** für die zusätzlich Relaiskarte zuständig. Es sind diverse Relaisausgänge für verschiedenste Anwendungen vorhanden. Die Kameras werden je nach Helligkeit zu- oder abgeschaltet. Die Bildauslösegeschwindigkeit ist abhängig ob ein definierter Magnetfeldalarm vorliegt oder nicht. Im Normalbetrieb bei Nacht erfolgt alle 2 Sekunden eine Bildauslösung. Bei erkanntem Alarm (1 oder 2) erfolgt alle 0,5 Sekunden eine Bildauslösung. Diese erfolgt 0,25 Sekunden zeitversetzt für zwei Kameras die möglicherweise parallel montiert sind. Damit entsteht eine gemeinsame Bildwiederholung beider Kameras zusammen mit 4 Bilder pro Sekunde. Das ist jedoch nicht mit jedem Kameratyp zu machen. Die Alpha 7s von Sony kann dies auch bei Nacht und ist aktuell die erste Wahl für solch eine Funktion.

Auf der 32GB SD-Karte im Raspberry Pi werden auch alle Videodateien und Meßwertdateien abgelegt. Diese können direkt mit einem externen Linuxsystem ausgelesen werden, ohne auf ein Netztwerk zurück zu greifen. Sehr einfach und Systemnahe wird ein Raspberry Pi 2 aufwärts mit Monitor, Maus und Tastatur benutzt. Die SD-Karte einschieben und dann Videos und Daten auf einen USB Stick laden. Änderungen an den Pythonprogrammen, bzw. Veränderungen von Zeiten bzw. Schwellenwerten sind damit auch möglich. Über Windows ist es sehr einfach mit **VNC und einem Netzwerk** die Videodateien und Meßwertdateien herunter zu laden. Dies gilt im besonderen für den Raspberry Pi Zero-W.

Das gesamte Betriebsystem mit allen notwendigen Programmen ist auf einer 32GB micro SD-Karte als ISO Image gebrannt. Für den einfachen Anwender ist solch eine SD-Karte die richtige Wahl. Möchte der Anwender eine andere Sensorkarte benutzen, dann sind die Python Programme diesbezüglich zu ändern. Das setzt jedoch einige Erfahrung im Umgang mit dem Raspberry Pi voraus. In den Videobildern sind die wichtigsten Informationen eingeblendet.

## 

Von links nach rechts:

Projektbezeichner, Datum, Zeit, Belichtungszeit des Einzelbilds, Betriebszustand, Magnetfelder in Raumrichtung X-Y-Z, Temperatur und Luftdruck.

Wird zusätzlich die von uns entwickelte Relaiskarte verwendet, stehen überwiegend potentialfreie Relaiskontakte zur Verfügung, die durch das **Programm Kamerasteuerung** bedient werden. Die Einbindung dieser Funktionen für externer Geräte basiert auf der Erfahrung von mehr als 10 Jahren praktischer UFO-Forschung mit technischen Geräten.

Die Funktionsabläufe sind speziell auf Kameras der Firma Sony abgestimmt. Für Anwender die andere Kameratypen bevorzugen ist es möglich das Phytenprogramm hierfür entsprechend ab zu ändern. Das Programm ist gut kommentiert.

Der Kameratyp HX60V wird von mir nur bei kleinen Anlagen verwendet. Diese Kamera muß jedoch in grober Art und Weise umgerüstet werden und ist daher nicht jedermann's Sache.

Die Alpha 7s wird nur mit einem kommerziellen 5V Akkuadapter ausgerüstet. Die Bildauslösung erfolgt mit einem kommerziellen USB – Fernauslöser – Kabel. Für jede Kamera ist eine eigene Powerbank vorgesehen die direkt über den jeweiligen Einschaltkontakt geschaltet wird.

In der Praxis sind dann in einem ausgebauten System (zB.Überwachungskoffer) 1 (2) Powerbanks enthalten. Für den(die) Raspberry(s) und für die Sony(s) Alpha 7s.

Die Relaiskontakte für Alarm , Schaltpunkt 1 und Schaltpunkt 2 können beliebig verwendet werden (Schaltpunkt 1+2 sind Lichtabhängig) , zB. für externe Videorecorder .....

Die zwei Digitaleingänge (14 und 15) werden zur manuellen Einstellung der Lichtempfindlichkeit bzw. zum Nacht-Dauerbetrieb benutzt. Der Umschaltpunkt Tag/Nacht erfolgt über die Lichtwerte der Raspi-Kamera. Ohne Brücke entsprechend der Maximalbelichtung durch die Raspikamera. Mit Brücke 15 etwa 50% früher. Mit Brücke 14 wird dieser Lichtmodus ausgeschaltet und die Relais arbeiten lichtunabhängig. Diese Einstellung ist eine Anpassung an verschieden externe Kameratypen und deren Lichtempfindlichkeit.

Mit einer Brücke werden die LEDs den Relais zugeschaltet (gedacht für Testzwecke).

Ein Steckplatz I<sup>2</sup>Bus für eine Hardwareuhr ist vorbereitet. Da der Raspberry Pi keine gepufferte Uhr hat, bleibt nach dem Ausschalten die Zeit stehen.

# Anschaltung von externe Geräten



Für die Erweiterungen durch zukünftiger Fremdmodule ist ein weiterer I<sup>2</sup>Bus Steckplatz vorgesehen.

Bei manchen Powerbank ist es notwendig diese über einen Impuls einzuschalten. Für solche Akkus ist die Dopppelfunktion HX60V/Powerbank vorgesehen. Zum besseren Verständnis der Schaltplan.



Ausbaumöglichkeiten mit verschiedenen Modulen

Es besteht die Möglichkeit die \_\_ITUF\_\_ Hardware mit unterschiedlichen Sensoren aus zu rüsten. Das aktuell fertig entwickelte System arbeitet mit dem Enviro pHAT von PIMORONI. Leider wurde diese Sensoreinheit vom Hersteller abgesagt und ist nur noch vereinzelt als Restposten zu erwerben.



Dieses System ist über Monate unter Praxisbedingungen getestet und arbeitet fehlerfrei. Die Wahl des Raspberry Pi ist beliebig. Auch der Zero ist möglich.

Wichtig ist, das die Sensoren (Flachbandkabel) nicht zu nahe an den Reed-Relais sind, da dies zu dauerhaftem Magnetfeldalarm führt.

Eine weiter Möglichkeit ist der Einsatz eines etwas begrenzten Sensors. Der LSM 303D ist auch von der Firma PIMORONI und in etwas abgewandelter Form im Enviro pHAT verbaut. Jedoch werden ausschließlich Magnetfeld und "Gravitation" erfaßt.

In dieser Kombination wird kein Flachbandkabel benötigt.

Dieses System wurde noch nicht getestet.

Möglicherweise ist es notwendig den Sensor mit einem 5-adrigen Flachbandkabel ab zu setzten. (Störungen durch die Reed-Relais).



Der Einsatz des LSM 303D Sensormoduls funktioniert nur, wenn die Programme messwerterfassung.py und videorecorder.py auf dieses Modul angepaßt werden.

Das ist kein Hexenwerk, jedoch nichts für Raspberry Pi –Anfänger. Begrenzte Kenntnisse von Python sind auch notwendig.

Die vorhandenen Python-Programme sind derart gut Kommentiert, das es eine lösbare Aufgabe ist.

Eine weiter Möglichkeit ist der Einsatz des Sense HAT Sensors. Diese Sensorplatte ist in Funktion und Ausbau mit der Enviro pHAT von PIMORONI vergleichbar.

Dieses System wurde noch nicht getestet.



Der Einsatz des Sense HAT Sensormoduls funktioniert nur, wenn die Programme messwerterfassung.py und videorecorder.py auf dieses Modul angepaßt werden.

Das ist kein Hexenwerk, jedoch nichts für Raspberry Pi –Anfänger. Begrenzte Kenntnisse von Python sind auch notwendig.

Die vorhandenen Python-Programme sind derart gut Kommentiert, das es eine lösbare Aufgabe ist.

## **Oberfläche Raspberry PI**

Über VNC oder direkt mit Monitor, Tastatur und Maus werden die Daten vom Raspberry Pi /Zero auf einen beliebigen PC oder einen USB-Stick kopiert.



Das ist der Eröffnungsbildschirm. Wenn man auf das Ordnersymbol klickt, gelangt man zu den relevanten Daten.



Hier sind die Pythonprogramme und Subprogramme die automatisch nach dem Einschalten aktiviert werden. Auch beim aktuellen Zugriff laufen diese im Hintergrund weiter. Dies beeinflußt die Datenübertragung unmerklich. An die Pythonprogramme sollte man nur Hand an legen, wenn man weis was man macht.



Das sind die Videodateien (Muster) die später auf einem beliebigen System geprüft werden.



Das sind die Meßwertdateien (Muster) die später auf einem beliebigen System geprüft werden.

Es ist sinnvoll, nach dem alle Daten gesichert sind, die auf dem Raspberry vorhandenen Daten in den Mülleimer zu werfen. Sie sind immer über den Speicherplatz informiert. Wird dieser zu klein, dann ist der Mülleimer zu leeren, da nichts überschrieben wird.

🐞 🌒 🖻 🖬 🌞 🔇	Digraspben	rypi: ~	erfassu	ng_daten	2	TUF_Ser	sor_20200	5		(V2)	LT 1	48	e s. 15	26	4
			IT.	JF_Senso	r_2020062	27_1301.tx	ť							- 11	×
Date Bearbeiten Suchen Opt	tionen Hilfe														
Datum/Zeit	Sekunden	A1	nax nav	NOZ	gravx	gravy	gravz	Temp	Druck	AB	A1	A2	A3	3	101
2020-06-27_13:01:44.213	1593255704.2	6 1	450 -477	- 3958	-0.032	-0.044	1.012	31.92	990.355	1.654	1.654	1.654	1.654	r i	1
2020-06-27 13:01:45.300	1593255705.3	6 1	433 -479	-3947	-0.024	-8,858	1.016	31,89	998.346	1.654	1.654	1.654	1.654	£	
2020-06-27 13:01:46.379	1593255706.4	6 1	460 -465	-3944	-0.033	-0.047	1.017	31.87	990.344	1.654	1.654	1.654	1.654	1	0
2020-06-27 13:01:47.457	1593255707.5	6 1	428 -476	-3947	-0.039	-8.844	1.014	31.84	998.336	1.654	1.654	1.654	1.654	£	
2020-06-27 13:01:48.534	1593255708.5	6 1	425 -475	- 3923	-0.025	-0.044	1.041	31.83	990.331	1.654	1.654	1.654	1.654	1	
2020-06-27 13:01:49.612	1593255709.6	8 1	453 -453	-3936	-0.029	-8,845	1.011	31.82	998.329	1.654	1.654	1.654	1.654	1	
2020-06-27_13:01:50.689	1593255710.7	0 1	440 -479	-3909	-0.030	-0.045	1.011	31.81	990.325	1.654	1.654	1.654	1.654		
2020-06-27 13:01:51.766	1593255711.8	6 1	461 -479	-3917	-0.034	-8.844	1.038	31.88	998.321	1.654	1.654	1.654	1.654	£	
2020-06-27_13:01:52.844	1593255712.8	6 1	432 -471	-3875	-0.028	-0.049	1.016	31.79	990.315	1.654	1.654	1.654	1.654	1	
2020-06-27_13:01:53.921	1593255713.9	8 1	420 -473	-3923	-0.035	-8.848	1.028	31.79	998.311	1.654	1.654	1.654	1.654	£	
2020-06-27_13:01:54.999	1593255715.0	6 1	449 -469	-3925	-0.027	-0.047	1.020	31.79	990.309	1.654	1.654	1.654	1.654	1	
2020-06-27 13:01:56.076	1593255716.1	8 1	418 -480	-3937	-0.026	-0.848	1.014	31.79	998.387	1.654	1.654	1.654	1.654	1	
2020-06-27_13:01:57.153	1593255717.2	6 1	438 -475	- 3927	-0.024	-0.050	1.022	31.79	990.307	1.654	1.654	1.654	1.654		
2020-06-27 13:01:58.231	1593255718.2	6 1	484 -479	-3920	-0.027	-0.045	1.013	31.79	998,364	1.654	1.654	1.654	1.654		
2020-06-27_13:01:59.308	1593255719.3	6 1	448 -469	-3933	-0.025	-0.045	1.011	31.79	990.303	1.654	1.654	1.654	1.654	1	
2020-06-27 13:02:00.386	1593255720.4	6 1	438 -472	-3943	-0.029	-8.842	1.019	31.80	998.383	1.654	1.654	1,654	1.654		
2020-06-27_13:02:01.463	1593255721.5	6 1	419 -461	-3921	-0.015	-0.046	1.016	31.80	990.300	1.654	1.654	1.654	1.654	£	
2020-06-27_13:02:02.547	1593255722.5	6 1	411 -435	-3927	-0.024	-8.849	1.008	31,80	998.382	1.654	1.654	1.654	1.654	£	
2020-06-27_13:02:03.625	1593255723.6	6 1	415 +485	-3943	-0.026	-0.043	1.024	31.81	990.301	1.654	1.654	1.654	1.654	1	
2020-06-27_13:02:04.703	1593255724.7	6 1	426 -466	-3929	-0.028	-8.846	1.019	31.81	998.296	1.654	1.654	1.654	1.654	£	
2020-06-27_13:02:05.780	1593255725.8	6 1	392 -468	-3897	-0.034	-0.045	1.008	31.82	990.294	1.654	1.654	1.654	1.654	1	
2020-06-27_13:02:06.858	1593255726.9	8 1	438 -456	-3938	-0.031	-8.853	1.024	31.83	998.295	1.654	1.654	1.654	1.654	1	
2020-06-27_13:02:07.950	1593255728.0	8 1	422 -458	-3926	-0.025	-0.047	1.021	31.83	990.294	1.654	1.654	1.654	1.654		
2020-06-27_13:02:09.027	1593255729.0	8 1	411 -493	-3919	-0.023	-8.847	1.020	31.84	998.294	1.654	1.654	1.654	1.654	£	
2020-06-27_13:02:10.105	1593255730.1	6 1	423 -468	-3943	-0.027	-0.844	1.020	31.85	990.287	1.654	1.654	1.654	1.654	1	
2020-06-27_13:02:11.182	1593255731.2	6 1	397 -453	-3965	-0.022	-8.844	1.008	31.85	998.283	1.654	1.654	1.654	1.654	£	
2020-06-27_13:02:12.259	1593255732.3	6 1	428 -445	-3943	-0.028	-0.043	1.025	31.86	990.281	1.654	1.654	1.654	1.654	1	
2828_86.27 19.82.19 997	1509255799 9	R 1	498 +487	. 90.20	_A A35	-R 845	1.818	21 87	008 278	1.654	1.654	1.654	1.65/		2

So sieht die Liste der Meßwerte aus, die mit einem beliebigen Programm (zB. Exel) in Meßkurven dargestellt werden können. In jeder Datei sind 1 000 Meßreihen enthalten.

Von links nach rechts:

Datum/Zeit,	Klartextanzeige bis in den Millisekunden Bereich.				
Sekunden,	Darstellung von Datum/Zeit nach internationaler Regel.				
Al,	Zeigt den Alamzustand als Ziffer 0-2				
Mgx-mgz,	Magnetfeldanzeige (hier wie vom ENVIR PHAT geliefert)				
Gravx-gravz,	Gravitationswerte (hier wie vom ENVIR PHAT geliefert)				
Temp,	Gibt die Temperatur auf dem ENVIR PHAT an.				
Druck,	Luftdruck (hier wie vom ENVIR PHAT geliefert).				
A0 bis A3,	Gibt die Werte der vier Analogeingänge wieder (ENVIR PHAT)				

Wird eine andere Meßkarte auf den Raspberry Pi aufgesteckt, muß das Pythonprogramm entsprechend angepaßt werden, sonst geht gar nichts.

### Wenn ein neues Image erstellt werden soll

Dies ist für reine Anwender etwas gewöhnungsbedürftig. Reine Anwender benutzen besser das vorhanden Image auf der SD-Karte. Es ist geplant das das Image auf einem Server abgelegt wird und für Jedermann abrufbar ist.

Raspberry ZeroW Setup

Benötigte Tools: Putty: <u>https://www.heise.de/download/product/putty-7016</u> Win32DiskImager: <u>https://www.heise.de/download/product/win32-disk-imager-92033</u> RealVNC: <u>https://www.realvnc.com/de/connect/download/vnc/</u>

Raspbian Stretch Download: <u>https://www.raspberrypi.org/downloads/raspbian/</u> >> Raspbian Stretch with Desktop

Imagedatei aus ZIP auspacken und mit Win32DiskImager auf die SD Karte schreiben.

Mit Windows Explorer die SD Karte öffnen und dort eine leere Textdatei anlegen mit dem Namen 'ssh'.

#### NICHT 'ssh.txt' sondern nur 'ssh' !!

(Explorer-Einstellungen ändern dass Dateiendungen angezeigt werden) Nur wenn die Datei 'ssh' vorhanden ist kann man per ssh (Putty) auf den Raspi zugreifen.

Auf der SD Karte eine Datei 'wpa\_supplicant.conf' anlegen mit folgenden Inhalt:

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
```

Dabei ist ssid der Name des WLANs und psk ist das WLAN Passwort.

Nun kann die SD Karte in den Raspi gesteckt und dieser eingeschaltet werden.

Nachdem er mit Booten fertig ist, sollte er im WLAN Router als 'RASPBERRYPI' erscheinen und es sollte auch die IP-Adresse angezeigt werden die er vom Router zugewiesen bekommen hat. Diese IP-Adresse bleibt in der Regel immer gleich weil der Router das Gerät anhand seiner MAC-Adresse später wiedererkennt. Zumindest solange man keine anderen Geräte neu anmeldet.

Putty starten und mit Rasip connecten (IP-Adresse aus dem Router verwenden):

Reputity Configuration	×
Category:	
<ul> <li>Session</li> <li>Logging</li> <li>Terminal</li> <li>Keyboard</li> <li>Bell</li> <li>Features</li> <li>Window</li> <li>Appearance</li> <li>Behaviour</li> <li>Translation</li> <li>Selection</li> <li>Colours</li> <li>Connection</li> <li>Data</li> <li>Proxy</li> <li>Telnet</li> <li>Rlogin</li> <li>SSH</li> </ul>	Basic options for your PuTTY session         Specify the destination you want to connect to         Host Name (or IP address)       Port         192.168.1.10       22         Connection type:       Raw         Raw       Telnet         Rlogin       SSH         Serial         Load, save or delete a stored session         Saved Sessions         Default Settings         Load         Quelter
Serial	Close window on e <u>x</u> it: ◎ Always ◎ Never
About	<u>Open</u> <u>Cancel</u>

Dieses Warnfenster mit 'Ja' bestätigen:

PuTTY Sec	urity Alert
	WARNING - POTENTIAL SECURITY BREACH! The server's host key does not match the one PuTTY has cached in the registry. This means that either the server administrator has changed the host key, or you have actually connected to another computer pretending to be the server. The new rsa2 key fingerprint is: ssh-rsa 2048 d4:b8:37:0f:35:27:c1:51:bf:a7:18:ba:18:7a:a1:27 If you were expecting this change and trust the new key, hit Yes to update PuTTY's cache and continue connecting. If you want to carry on connecting but without updating the cache, hit No. If you want to abandon the connection completely, hit Cancel. Hitting Cancel is the ONLY guaranteed safe choice.
	Ja <u>N</u> ein Abbrechen

Nun als 'pi' anmelden mit Passwort 'raspberry':



Das Passwort sollte nun geändert werden mit 'passwd'.

Wichtige Einstellungen mit raspi-config:

#### sudo raspi-config

bringt dieses Menü:



Diese Einstellungen sollten auf jeden Fall gemacht werden:

**3 Boot Options** 

B1 Desktop / CLI B4 auswählen

4 Localisation Options

I2 Change Timezone Europe / Berlin auswählen

**5** Interfacing Options

P1 Camera >> Enable P3 VNC >> Enable P4 SPI >> Enable P5 I2C >> Enable

7 Advanced Options

A1 Expand Filesystem

A5 Resolution >> gewünschte Desktop Auflösung auswählen

raspi-config beenden mit <Finish> nun erfolgt ein reboot.

### Autostart anlegen

so geht das mit dem Autostart (ist gleich das erste von 214000 Google-Ergebnissen): https://raspberry.tips/raspberrypi-einsteiger/raspberry-pi-autostart-von-skripten-und-programmeneinrichten

Am einfachsten ist die Methode mit rc.local, d.h.:

-----

In der Konsole: (für Autostart)

sudo nano /etc/rc.local "Das ist der Aufruf zum Editor"

dann in Editor über Pfeiltasten nach ganz unten gehen und dann eingeben:

python /home/pi/Pythonprogrammname exit 0

dann Controll X dann antworten mit Y

nun erscheint Datenmeldung von Raspi

Starten ohne Autostart, dann in Kosole eingeben:

python /home/pi/Pythonprogrammname.py

Enviro-Karte

Daten für

In Konsole eingeben: curl https://get.pimoroni.com/envirophat | bash

Das installiert die Software für den Enviro PHat.

Echtzeituhr DS3231 in das System einbinden:

sudo raspi-config dort Interfacing Options auswählen I2C auswählen ARM i2C Interface aktivieren

sudo nano /etc/rc.local folgendes Befehle in die rc.local einfügen:

echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new\_device hwclock --hctosys systemctl restart systemd-timesyncd.service sleep 5 hwclock --set --date="\$(date "+%m/%d/%y %H:%M:%S")"

sudo nano /etc/modules folgendes Befehle in die modules einfügen:

i2c-dev i2c-bcm2708 rtc-ds1307

Wird diese Änderung gemacht, dann unbedingt das Uhrenmodul verwenden. Wird ein Image von uns verwendet, dann ist die Uhrenfunktion bereits enthalten. Wenn das Uhrenmodul nicht vorhanden ist, kann es zu Programmstörungen kommen. (Keine Aufzeichnung der Videos).

### Kameratest um die Brennweite mechanisch zu justieren:

Das geht nur bei einem Testaufbau ohne den automatischen Start der Pythonprogramme.

sudo nano /etc/rc.local
dann die Startdatei mit # auskommentieren
#python /home/pi/cam\_05.py
Programm neu starten.

In die Konsole eingeben: Raspistill -t 0

Nun erscheint das Echtzeit-Videobild.

Wenn die Kamera justiert ist, den ursprünglichen Eintrag herstellen.

sudo nano /etc/rc.local dann den Autostartbefehl wieder aktivieren python /home/pi/cam\_05.py

Programm neu starten.