

Pi-Kamera



Raspberry-Kamera-Aufbau 2023 - © by Dieter Broidhagen 51145 800n

Zutaten



Speicherkarte 256 G | USB-Netzteil | Raspberry Pi Zero-W | Kameramodul
Flachband-Flex-Kabel | USB-Kabel mit Stecker USB auf USB-C | Adapter Mico-USB auf USB-C
Ständer für Kamera | Bastel-Pappe | Bastelpapier | Krepp-Band und Tesafilm

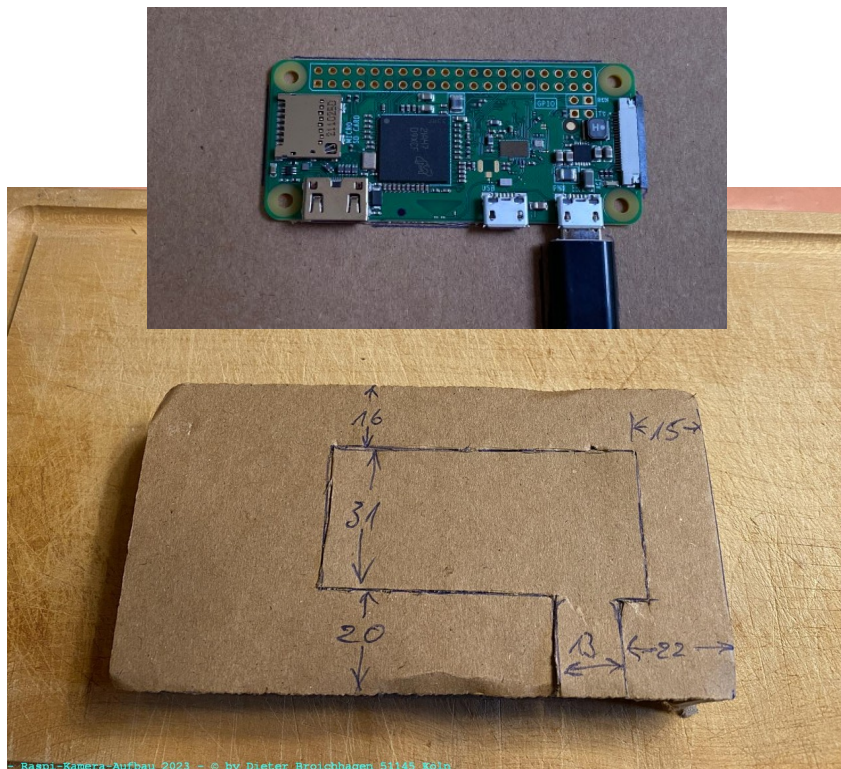


Die Bastelpappe ist 15 mm dick mit Wabenmuster im Inneren, war bei der Lieferung von einer Elektroheizung als Transportschutz vorgesehen.

Gehäuseträger für den Raspi Zero-W und Kamera-Modul

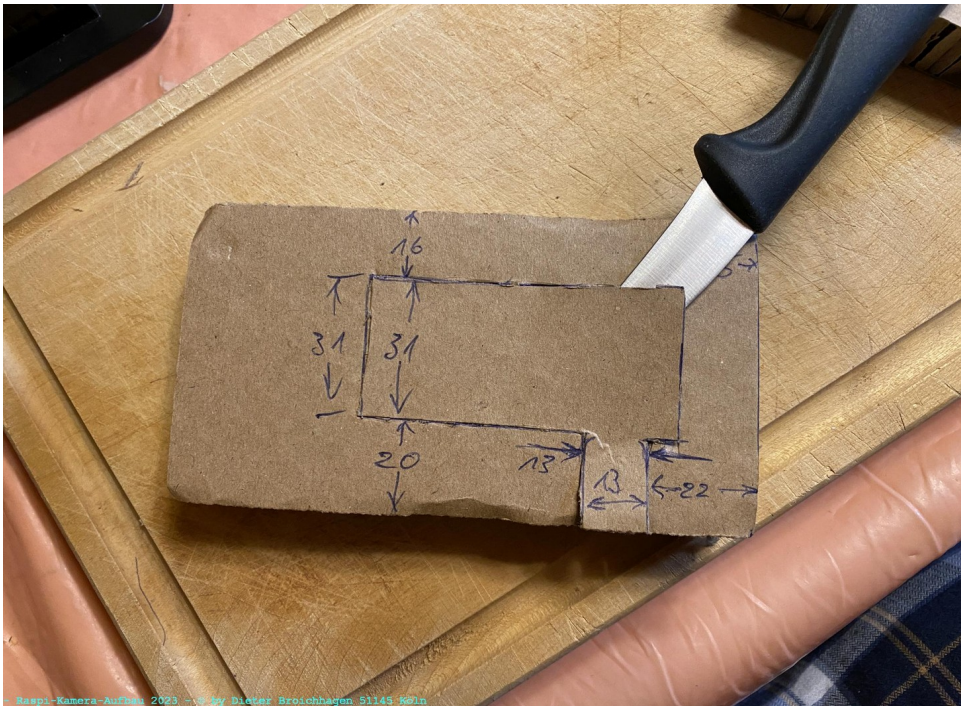


Unterteil für die Platine vorbereiten

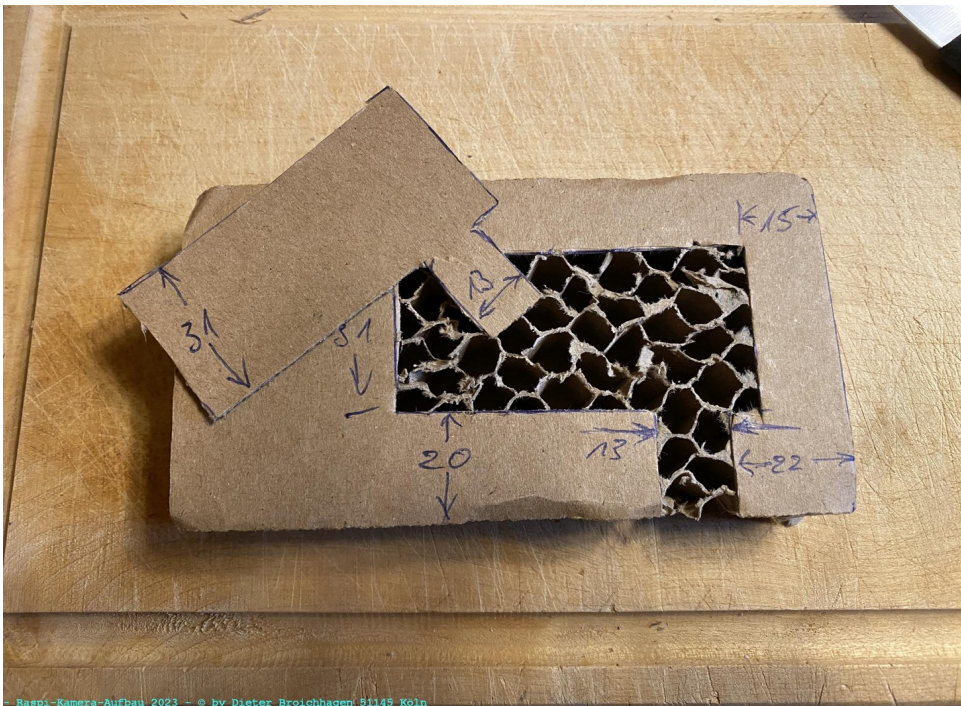


Pappe leicht einritzen

Oberer Teil vorsichtig entfernen



Die darunter liegenden Waben leicht eindrücken



Oberteil für die Platine vorbereiten



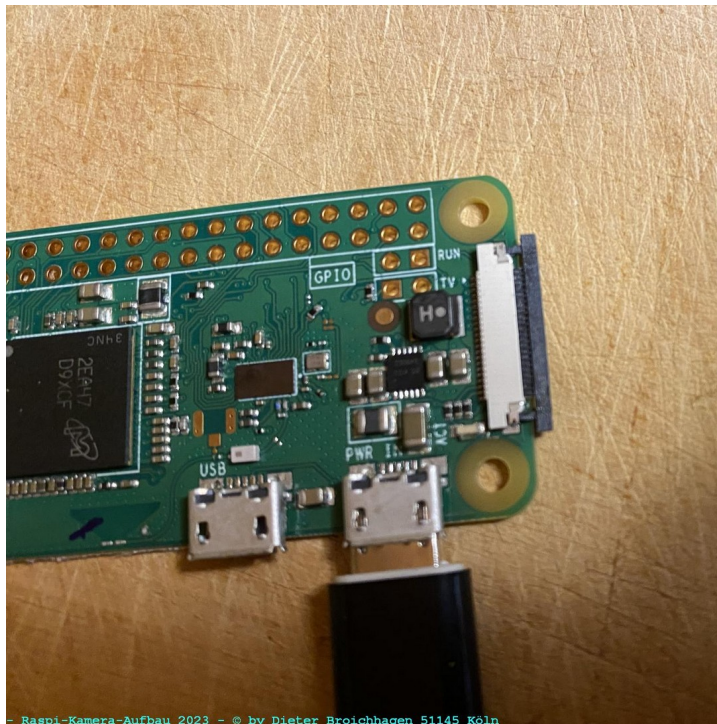
Das Papier können wir dann als Schablone für das Oberteil nutzen

Oberteil und Unterteil sind fertig



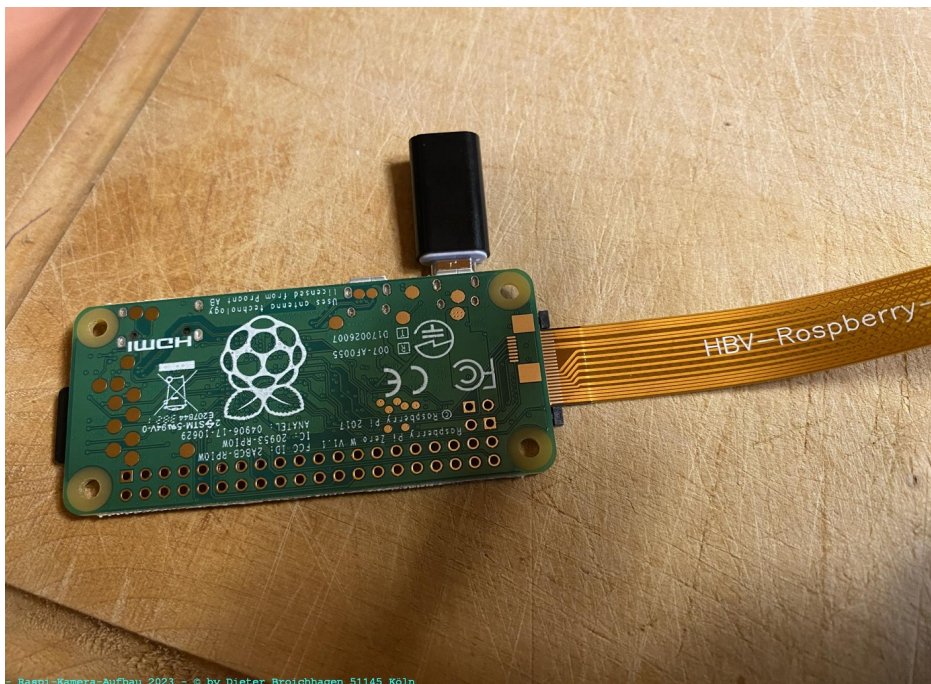
Papier entfernen wie beim Unterteil

Platine für Kamera-Modul vorbereiten

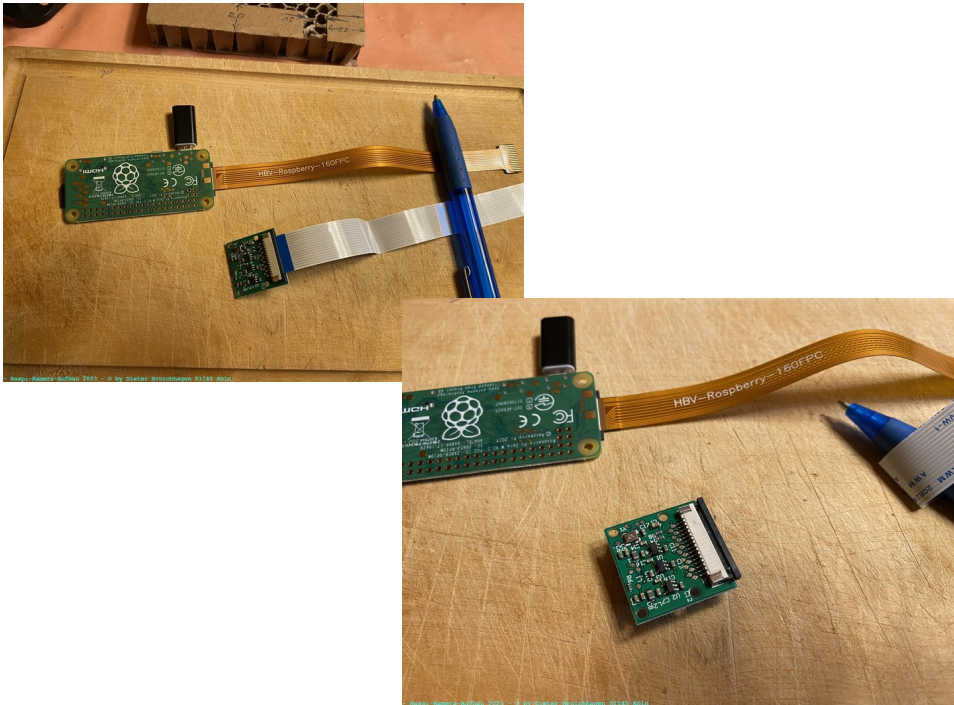


Pfostenstecker vorsichtig raus ziehen ca. 2 mm

Flachband-Flex-Kabel vorsichtig einführen und Pfostenstecker andrücken

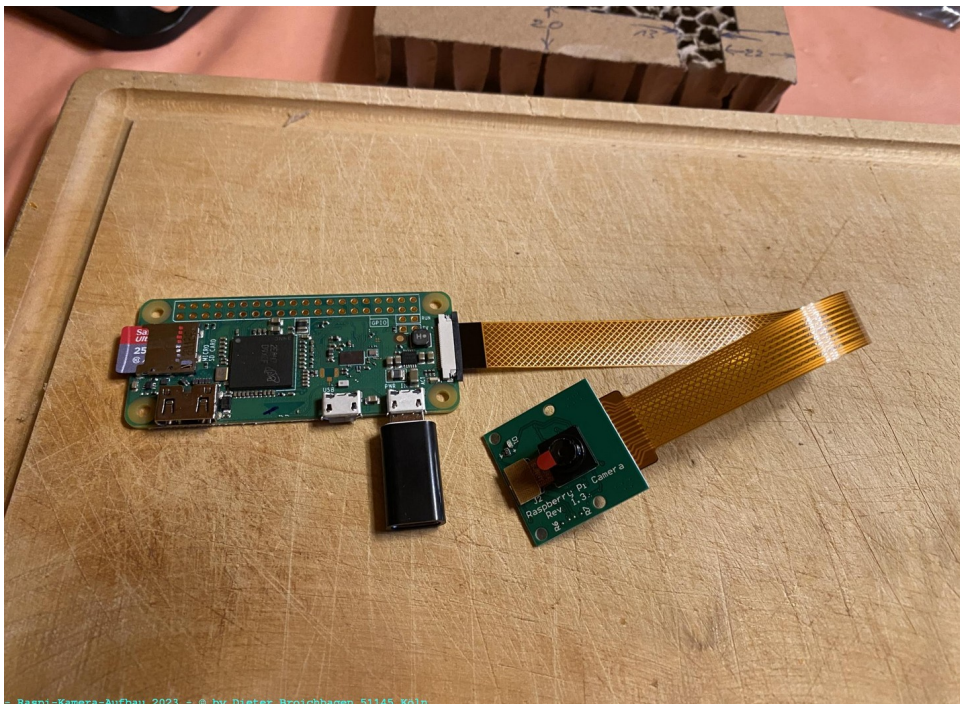


Das weiße Flachband-Flex-Kabel am Kamera-Modul entfernen

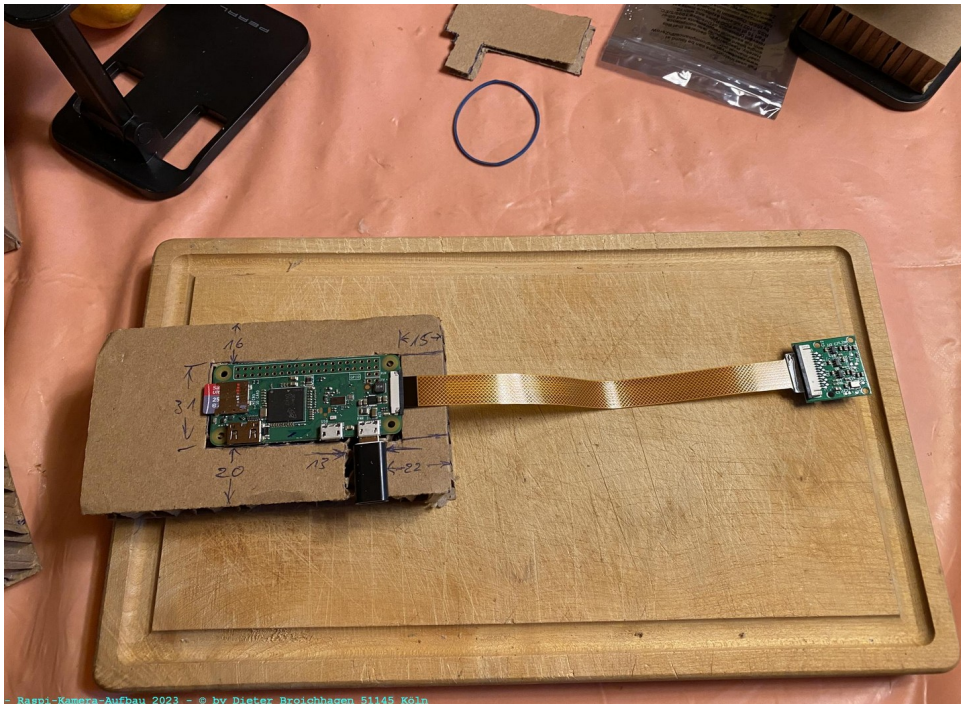


Das weiße Flex-Kabel passt nicht am Zero-W. Daher müssen wir es tauschen mit dem anderen Flex-Kabel, dass die richtigen Anschlüsse hat.

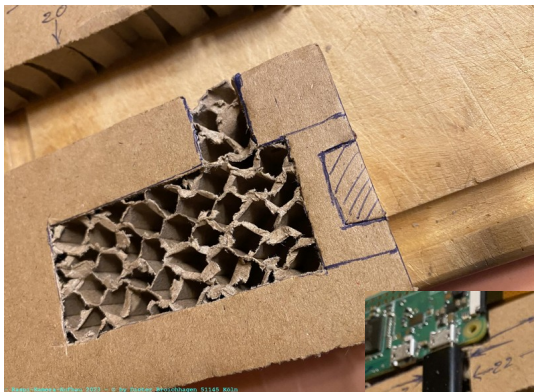
Kamera-Modul am neuen Flachband-Flex-Kabel anschließen



Platine auf das Gehäuseunterteil legen, nichts ankleben oder sonst wie fixieren!

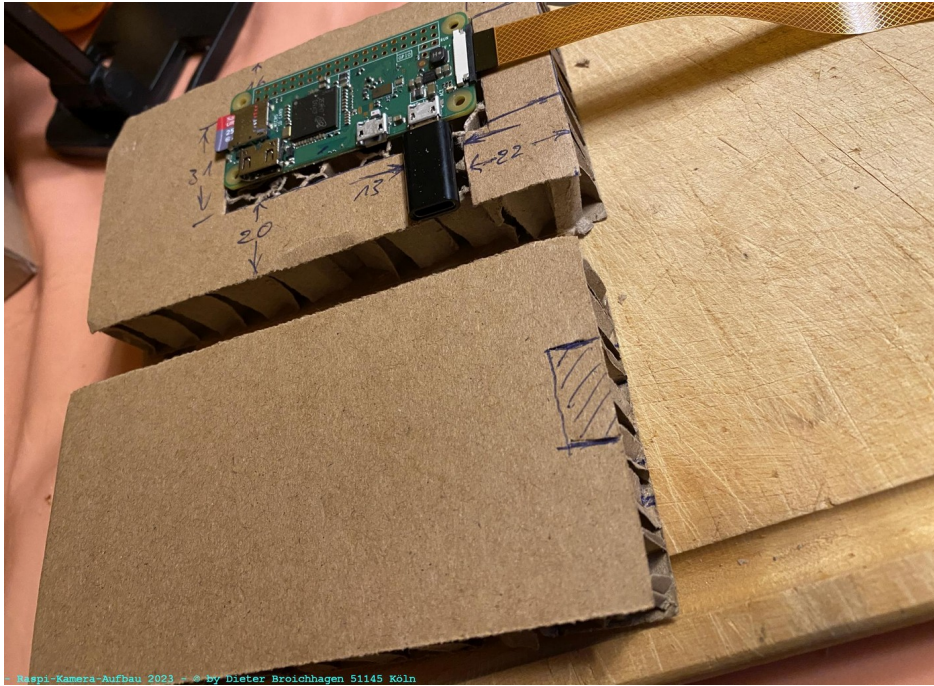


Oberteil zusätzlich vorbereiten



Wir brauchen einen Ausschnitt im Oberteil für das Flachband-Flex-Kabel, das komplett um 180 Grad umgeklappt wird.

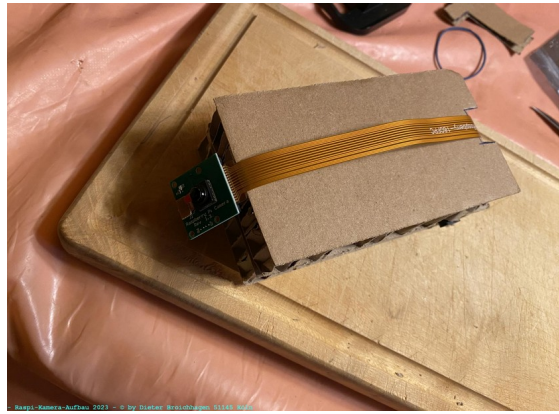
Unterteil mit Platine und Oberteil



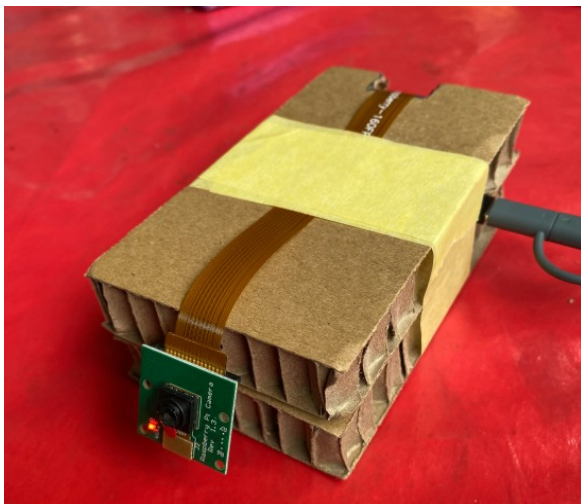
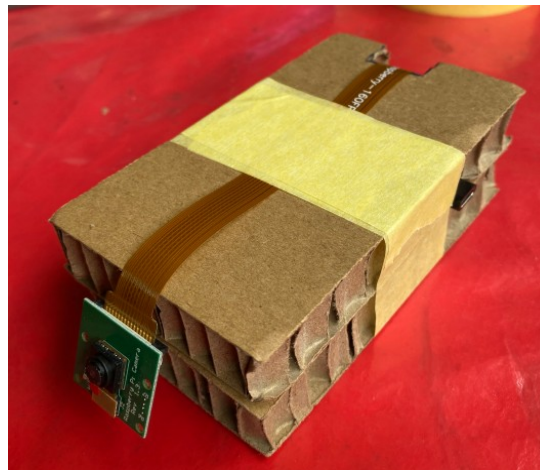
Oberteil – Ausschnitt für Flex-Kabel vorbereiten



Oberteil auf die Platine legen und Flex-Kabel um 180 Grad umklappen



Flex-Kabel mit Krepp-Band am Oberteil fixieren



Jetzt sollten wir die Kamera kurz testen ob sie funktioniert. Dafür schließen wir das USB-Kabel an das USB-Netzteil sowie an den Raspi. Es dauert ca. 2 Minuten bis das System hochgefahren ist. Wenn die Kamera Betriebsbereit ist, leuchtet die rote LED. Im Router erfahren wir seine IP-Adresse, die wir dann auf – Immer die gleiche Ip-Adresse verwenden – setzen. Wenn wir in der motion.conf den „stream_port auf 8081“ gesetzt haben, dann können wir uns nun im Webbrowser das Kamerabild ansehen, indem wir die IP-Adresse des Raspi eingeben gefolgt mit :8081. Beispiel: <http://192.168.2.16:8081>

Des weiteren sollten wir den ssh-Zugang testen ob er funktioniert.

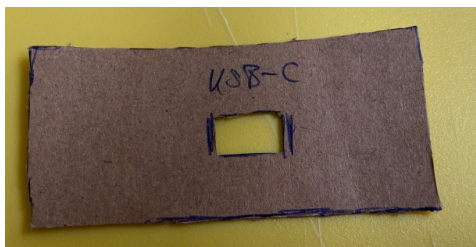
Wenn alles ok ist, können wir das USB-Kabel wieder abziehen und die Kamera zu Ende bauen.

Es ist wichtig das zu testen, da wir ja keinen Anschluss an den HDMI-Port nach außen führen.

Wenn die Kamera funktioniert, dann brauchen wir den HDMI-Port nicht.

Wenn später das System mal nicht mehr startet, müssen wir alles wieder zerlegen um notfalls einen neuen Speicherchip einzubauen.

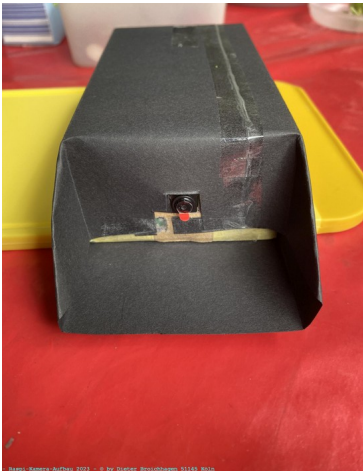
Gehäuse rundherum verstärken



Gehäuse rundherum mit Krepp-Band zukleben



Gehäuse mit Bastelpappe einpacken (wie bei Geschenkpapier :-))



Kamera mit Gummi auf den Ständer fixieren. Fertig ist die Kamera. :-))



Copyright by:

Dieter Broichhagen | Humboldtstrasse 106 | 51145 Köln | Web: <http://www.broichhagen.eu> | Mail: info@broichhagen.de
Es ist ausdrücklich gestattet, die Kamera in der zuvor beschriebenen Form nachzubauen.

Köln, den 25. Oktober 2023

Einrichtung und Vorbereitungen für eine neue Kamera mit dem Raspberry Pi Zero-W

Bauteile

- Raspberry Pi Zero-W
- microSDXC-UHS-I Speicherkarte 256 GB
- Kameramodul
- Flachbandkabel
- Adapter Micro-USB-Stecker auf USB-C-Buchse
- Kabel USB-Stecker auf USB-C-Stecker
- Netzteil 2 A mit USB-Buchse
- Ständer für die Kamera (von Pearl)

Alles zusammen bei Amazon gekauft für etwa 50 Euro

Ich habe nach dem Bau des Kamera-Prototyps von der Speicherkarte ein Backup erzeugt mit dem Konsolen-Tool "fsarchiver".
Damit kann ich nun jederzeit die Speicher-Karte klonen.

Die Partitionen auf der Speicherkarte habe ich wie folgt aufgeteilt:

1. Partition 128 MB mit Dateisystem Fat32 für den Boot-Sektor
2. Partition 64 GB mit Dateisystem ext4 für das Betriebssystem
3. Partition 174,17 GB mit Dateisystem ext4 für die Daten

Achtung!!! Für nachfolgende Befehle darf die Speicherkarte nicht gemountet sein!

Beispiel-Syntax mit fsarchiver Backup erzeugen

```
fsarchiver -v savefs /media/daten/daten/Backup/raspis/raspi-zero-w/raspi-zero-w-perfekt_2023-07-23.fsa /dev/mmcbk0p2
```

Beispiel-Syntax mit fsarchiver Backup zurückspielen

```
fsarchiver -v restfs /media/daten/daten/Backup/raspis/raspi-zero-w/raspi-zero-w-perfekt_2023-07-23.fsa id=0,dest=/dev/mmcbk0p2
```

Nach dem Zurückspielen des Backups auf eine neue Speicherkarte ist das Betriebssystem noch nicht lauffähig.

Wir benötigen die PARTUUID, die wir über den Konsolenbefehl blkid mit root-Rechte ermitteln.

Beispiel:

```
root@mint21:/home/dieter# blkid
```

..... ich habe hier nur die relevanten Partitionen aufgeführt

```
/dev/mmcbk0p1: LABEL_FATBOOT="BOOT" LABEL="BOOT" UUID="1F91-66DE" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="116ac27a-01"
/dev/mmcbk0p2: LABEL="rootfs" UUID="c68b37be-930e-49c8-87e8-9005e8247103" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="116ac27a-02"
/dev/mmcbk0p3: LABEL="daten" UUID="743c07f7-c9b2-471b-85cb-92c59cf1a8e6" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="116ac27a-03"
```

Wir benötigen die PARTUUID von /dev/mmcbk0p2. Die lautet hier PARTUUID="116ac27a-02".
Diese tragen wir nun in der Datei cmdline.txt ein die sich im Verzeichnis /boot befindet.

Von diesem Verzeichnis /boot habe ich übrigens auch eine Sicherung angelegt zusammen mit dem Betriebssystem.
Diese Dateien im Verzeichnis /boot müssen wir nun zurückspielen, indem wir diese einfach in die gemountete Partition /dev/mmcbk0p1 kopieren.

Dann müssen wir in der Datei /etc/fstab ebenfalls die PARTUUID von allen Partitionen anpassen.

Wir müssen nun noch zwei Verzeichnisse anlegen. In der Partition /dev/mmcbk0p3 legen wir mit dem Konsolen-Befehl mkdir ein Verzeichnis daten an und in diesem Verzeichnis dann ein Verzeichnis picam. Die Zugriffsrechte setzten wir für alle auf lesen/schreiben/ausführen.

Ich habe im Verzeichnis /home/dieter einen Link zu diesem eben angelegten Verzeichnis picam angelegt. Somit wandern alle Kameraaufnahmen auf die Daten-Partition.

Nun tragen wir in der Datei /etc/hostname einen neuen Namen ein, z.B.: zero-w-07 und in der Datei /etc/hosts tragen wir neben 127.0.1.1 zero-w-07 ein.

Jetzt müssen wir noch in die Datei /etc/motion/motion.conf einige Zeilen anpassen.

```
text_left CAMERA-07      (wird auf den Aufnahmen links unten eingeblendet)
stream_port 8081         (falls dies erwünscht ist)
minimum_frame_time 1     (das Live-Bild ist dann etwas besser als bei 5)
target_dir /home/dieter/picam (das Verzeichnis für die Ablage der Aufnahmen)
```

Link zu Einrichtung von „motion“: <https://raspberry-valley.azurewebsites.net/Streaming-Video-with-Motion/>

Werden Änderungen an der o.a. Datei vorgenommen muss man anschließend das System neu starten oder den Dienst von Motion neu starten, damit die Änderungen wirksam werden.

Stand: 24. Oktober 2023 / Broi