

[Home \(https://www.pcwelt.de/\)](https://www.pcwelt.de/) > [Computer & Technik \(https://www.pcwelt.de/computer-technik/\)](https://www.pcwelt.de/computer-technik/) > [PC & Peripherie \(https://www.pcwelt.de/computer-technik/pc-peripherie/\)](https://www.pcwelt.de/computer-technik/pc-peripherie/) > [PC-WELT Hacks \(https://www.pcwelt.de/computer-technik/pc-peripherie/pc-welt-hacks/\)](https://www.pcwelt.de/computer-technik/pc-peripherie/pc-welt-hacks/)

# Mit Raspberry Pi Zero und Pi Cam unterwegs

08.03.2018 | 14:15 Uhr | Swen Hopfe ()



Swen Hopfe

Swen Hopfe arbeitet bei einem mittelständischen Unternehmen aus der Chipkarten-Branche und ist Experte für Smart Cards, RFID, das IoT, Raspberry Pi und Arduino. [Mehr \(/autoren/swen\\_hopfe\\_1523011113498823183\)](#)

Autorenprofil schließen ()

**Infrarot-Fotografie ist unser Hobby. Wir haben eine [mobile \(http://www.mobile.de\)](http://www.mobile.de) Pi Cam gebaut, mit der wir bei Tageslicht sowohl Effekt-Aufnahmen mit Infrarotfilter als auch ganz normale Aufnahmen machen können.**



Mittels Raspberry Pi und Pi Cam haben wir schon einige Vorhaben realisiert: eine [Scan-Vorrichtung \(https://www.pcwelt.de/a/raspberry-pi-und-pi-cam-als-scanner-verwenden,3446689\)](https://www.pcwelt.de/a/raspberry-pi-und-pi-cam-als-scanner-verwenden,3446689), eine [Webcam \(https://www.pcwelt.de/a/webcam-mit-raspberry-zero-pi-w-einrichten,3448168\)](https://www.pcwelt.de/a/webcam-mit-raspberry-zero-pi-w-einrichten,3448168) und eine [Fotofalle zur Tierbeobachtung \(https://www.pcwelt.de/a/naechtliche-tierfotografie-mit-noir-cam-und-raspberry-pi,3447354\)](https://www.pcwelt.de/a/naechtliche-tierfotografie-mit-noir-cam-und-raspberry-pi,3447354). Letztere ist schon erneuert worden und auch unsere Webcam gibt es schon in der zweiten Generation mit dem Raspberry Pi Zero. Der eignet sich gut im Zusammenspiel mit den originalen Kameramodulen und braucht wenig Platz. Da lag es nahe, mit dem Zero eine kompakte Kamera aufzubauen, die man auch unterwegs gebrauchen kann.

...utage hat zwar jeder ein Handy dabei und kann Schnappschüsse machen, aber da ist ja noch unser Hobby der Infrarot-Fotografie. Die IR-Variante, also die [NoIR-Cam \(https://www.pcwelt.de/a/naechtliche-](https://www.pcwelt.de/a/naechtliche-)

[tierfotografie-mit-noir-cam-und-raspberry-pi,3447354](#)) für den Pi, ist für den Hobbyfotografen nach wie vor interessant, da sie den Infrarotanteil des Tageslichts für den Bildsensor nicht sperrt, sondern ungedämpft durchlässt.


Ohne Umbau kann man in dieser Art und Weise nur noch ältere Digitalkameras zufriedenstellend nutzen. Für den Außeneinsatz eines Pi gab es bei uns bisher nur ein Provisorium, den großen Raspi im Kartongehäuse - relativ klobig und ohne großen Bedienkomfort. Neue Ideen hatten wir auch für die Stromversorgung und eine ordentliche Filterhalterung.

Also aufgemacht, um auch unsere [mobile \(http://www.mobile.de\)](http://www.mobile.de) Variante auf einen neuen Stand zu bringen und dort gleich den kleinen „Zero“ einzusetzen. Sie sollte uns ermöglichen, bei Tageslicht sowohl Effekt-Aufnahmen mit Infrarotfilter als auch ganz normale Aufnahmen zu machen. Und man sollte sie gut in der Hand halten oder abstellen können. Da wir sie nicht in der Nacht oder etwa unbeaufsichtigt nutzen wollen, brauchen wir kein spezielles wasserdichtes Gehäuse.

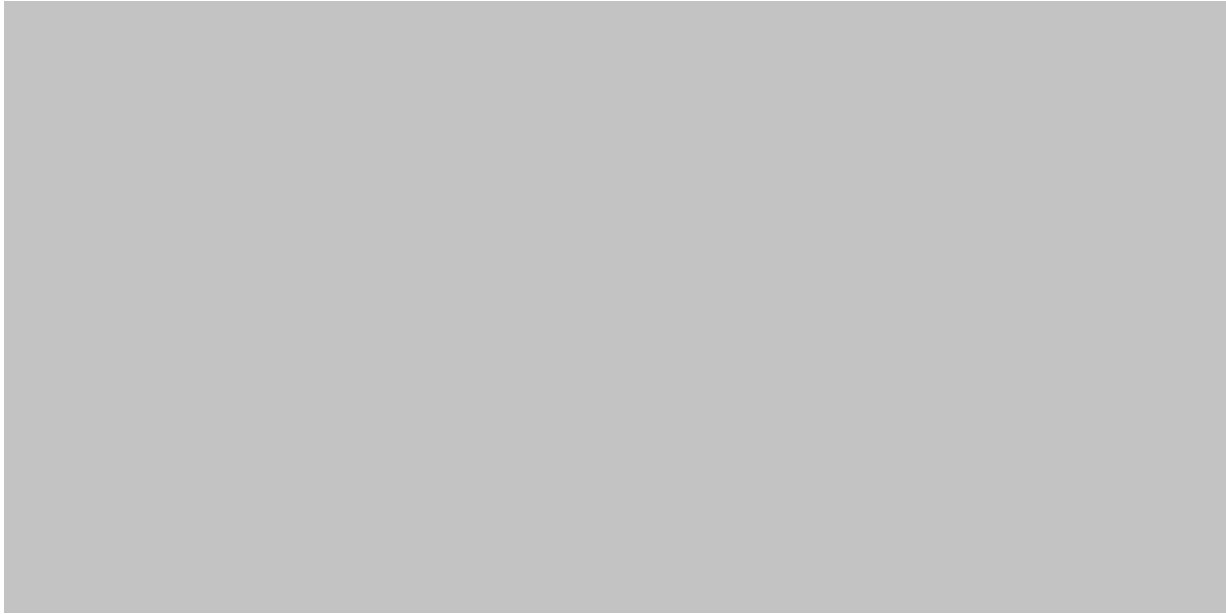
So kommt unsere neue „Zero Cam“ mit einem Touchpad, Echtzeituhr und OLED-Display daher. Die einzelnen Module sind auf einer Konsole mit integriertem Akku angebracht. Mittlerweile findet man auch Multifunktions-Kameramodule für den Pi, wo bei Bedarf ein Infrarotfilter vor dem Objektiv eingeschwenkt wird. Die funktionieren mitunter aber nicht zufriedenstellend, oft wird über Probleme mit der Mechanik berichtet. Außerdem ist so etwas nur die halbe Miete, denn man tauscht damit nicht die ganze Kamera, sondern nur den Filtervorsatz und muss somit immer mit der NoIR-Cam arbeiten. Für normale Aufnahmen ist die aber nicht geeignet, da sie farbverfälschte Bilder hervorbringt.

Bei unserem Aufbau lässt sich deshalb der komplette Kopf an der Front wechseln, wodurch die ganze Hardware (Pi und Cam) ausgetauscht wird. Das erschien uns als die optimale Lösung, um sowohl gute Aufnahmen unter Normallicht (Kopf mit normaler Pi Cam) als auch im Effekt-Modus (Kopf mit NoIR-Cam und Infrarotfilter) machen zu können.





Grundlage für unser Projekt ist der Raspberry Pi Zero W. Eine Beschreibung zur Inbetriebnahme möchte ich mir an dieser Stelle sparen, denn mittlerweile gibt es einige Anleitungen im Web dazu. Gut, dass der Raspberry eine große Community hat, in der viele längst auch ihre Erfahrungen mit dem kleinen Zero teilen. Durch seine Größe und integriertem WiFi bleibt auch bei uns alles schön kompakt. Dazu trägt auch die Verfügbarkeit von diversen Pi-Hats bei, die man gut stapeln kann.



Auf diese Weise haben wir auch eine RTC-Echtzeituhr und ein Touchpanel angebunden. Der „Touchphat“ bietet sechs beleuchtbare Eingabeflächen. Die LEDs lassen sich gezielt einzeln ansteuern. Es gibt gute Python-Unterstützung dafür. Günstig, wenn man wie wir auch das Steuerskript auf dem Pi in Python aufgebaut hat. Die Installation des „Touchphat“ geschieht unter Raspbian mit

```
$ sudo apt-get install python-touchphat
```

oder in Python3 mit

```
$ sudo apt-get install python3-touchphat
```

Wer sich zuerst anhand von Beispielen einarbeiten möchte, kann das Git-Archiv von Pimoroni klonen und in den Verzeichnissen stöbern:

```
$ git clone \
```

<https://github.com/pimoroni/touch-phat.git> (<https://github.com/pimoroni/touch-phat.git>)

Das RTC-Modul hat bei uns die Aufgabe, den Dateinamen der aufgenommenen Bilder einen Zeitstempel zu verpassen. Denn draußen im Feld hat der Pi keine Internetverbindung, wo er sich nach dem Hochfahren mit einem Zeitserver abgleichen könnte. In Kombination mit einem fortlaufenden Index steht für die aktuelle Aufnahme somit immer ein eindeutiger Dateiname zur Verfügung. Und es werden keine Bilder überschrieben, wie man es beim käuflichen Fotoapparat kennt.

Startet man unsere Kamera im Einzugsbereich vom heimischen WLAN, synchronisiert sich die Echtzeituhr automatisch auf die aktuelle „Netzzeit“ neu. In die Metadaten der Bilder kommt derzeit noch nichts, das wäre aber eine gute Erweiterung für die Zukunft und nicht so schwierig zu implementieren.



Den vorhin beschriebenen „Kopf“ unserer Zero Cam bildet das bekannte Gehäuseset mit „Himbeere-Logo“, wo man zwischen mehreren Varianten mit verschiedenen Aussparungen auswählen kann. Wir brauchen auf der Rückseite eine Öffnung für die GPIO-Leiste und an der Front das kreisrunde Loch für das Objektiv und wählen die zwei entsprechenden Schalen aus. Im Set ist auch ein kurzes Flachkabel mit der Verjüngung auf den kleineren Kameraport des Zero dabei.



Da es kleine Abweichungen zwischen den Revisionen 1.3 und 2.0 gibt, passen in die gelochte Oberschale des Sets nur Pi Cams der neuen Generation richtig rein. Die älteren Pi Cams ecken an, weil da rund um den Bildsensor das schwarze Plastik anders geformt ist. Das stört uns aber nicht, denn wir bohren den Deckel für den späteren Filteraufsatz großzügig auf.

Zwei „Köpfe“ haben wir uns nun aufgebaut, jeweils mit einem Pi Zero, einer normalen Pi Cam und einer NoIR-Cam. Am besten arbeitet man mit zwei Micro-SD-Karten. Beide Pi sind dann am Router als eigenständige Geräte angemeldet und können wechselseitig verwendet werden. Vor dem Schließen der Gehäuse kann man einen kurzen Test der Cams wie gewohnt mittels

```
$ sudo raspistill -o test.jpg
```

machen. Ist der Pi schon im lokalen Netzwerk, funktioniert das auch wie üblich in einem Terminal auf einem entfernten Desktop, wo man sich vorher mittels *ssh* verbunden hat. Hat man auf dem Raspi *feh* installiert, dann kann man *ssh* mit der *X*-Option aufrufen und sich das durch *raspistill* geschossene Bild gleich anzeigen lassen:

```
$ sudo raspistill -w 1200 -h 800 -o test.jpg ; feh --scale-down test.jpg
```

Ein bequemer Weg, denn es braucht dadurch weder einen X-Desktop auf dem Raspi noch eine Dateiübertragung, um das Aufnahmeergebnis grafisch sehen zu können.



Nach hinten schauen die Anschlüsse des Zero mittels Stiftleiste. Damit wird an den Rumpf der Kamera angedockt, also an die oben beschriebenen Module von RTC und Touchpanel. Die sind Teil vom restlichen Gehäuse. Im Fuß sind Lade-Elektronik und Akku untergebracht. Zum Laden ist eine Micro-USB-Buchse nach außen geführt.

In der Ansicht auf die Unterseite fehlt im folgenden Bild noch die Bodenplatte. In ihr haben wir zwei Aussparungen vorgesehen, damit man die Status-LEDs der Ladeplatine von außen sehen kann. Das war keine eigene Entwicklung, sondern ein Fertigkauf. Heutzutage findet man recht gute Elektronik, die als Lader für einen 3,7-V-Lipo-Akku Strom von einem handelsüblichen 5-V-Netzteil nehmen kann und gleichzeitig die Versorgung für den Verbraucher managt.

**Tipp** : [Raspberry Pi 3 als Desktopersatz verwenden \(https://www.pcwelt.de/ratgeber/Raspberry\\_Pi\\_als\\_Desktop\\_und\\_Netbook\\_nutzen-Mobiler\\_Eigen-PC-8674813.html\)](https://www.pcwelt.de/ratgeber/Raspberry_Pi_als_Desktop_und_Netbook_nutzen-Mobiler_Eigen-PC-8674813.html)





Bei uns sieht man also über diverse LEDs durch die Bodenplatte, ob die Betriebsspannung noch in Ordnung ist, ob gerade geladen wird oder ob etwa schon voll geladen ist. Mit 2000 mAh kommt man gut über den Tag und kann jede Menge Fotos machen. Damit sind schon fast alle Komponenten beschrieben. In der Vorderansicht sieht unsere Kamera so aus:



An der Front sieht man noch die weiße Oberschale des originalen Zero-Gehäuses. Zusätzlich zum Klick-Mechanismus ist sie für den Außeneinsatz durch zwei Hutmuttern gesichert. Der Rest ist schwarz lackiert worden. Löst man oben den geknickten USB-Stecker zur Stromversorgung, kann man den Kamerakopf inklusive Pi an der GPIO-Verbindungsleiste komplett abziehen und die Kameraversion wechseln. Dahinter sind das RTC- und Touchmodul angesteckt.

Der Kamera haben wir absichtlich einen breiten Fuß spendiert, weil dadurch genügend Platz für die Stromversorgung und diverse Buchsen entstehen konnte, nebst Montagefläche für das Display. Außerdem kann man das Ganze nun ordentlich auf einer ebenen Unterlage aufstellen.

Der Normalbetrieb funktioniert nun, aber der Kopf mit NoIR-Cam braucht noch einen Objektivaufsatz für den Infrarotfilter.





Deshalb wurde in die erweiterte Gehäuseöffnung noch ein Innengewinde geschnitten, in das ein kurzer Tubus mit dem Filter eingedreht werden kann. Darauf sitzt noch eine Schutzkappe. Die benötigten Teile kamen von einem Handy-Aufrüstset. Da ist das Filterglas natürlich nicht dabei und wir mussten einen käuflichen IR-Filter zerstören. Wir haben jedenfalls keine Miniaturvariante eines Filters von Hoya oder [Panasonic \(/handover/524?ws=1\)](#) gefunden. Das sind Filter, die ein Schraubgewinde für die gängigen Objektivgrößen von professionellen Kameras haben. Mit etwas Glück gewinnt man vom kaputten IR-Filter aber eine kleine Scherbe, die in den kleinen Durchmesser hineinpasst und eingeklebt werden kann.

Unsere anderen Kameras auf Basis Raspberry Pi sind stationär und liefern ihre Bilder mehr oder weniger automatisch. Da braucht es wenig Anzeige- und Bedienelemente am Gehäuse. Für unsere [mobile \(http://www.mobile.de\)](#) Kamera sollte es da schon etwas mehr Komfort sein, deshalb haben wir noch ein OLED zur Informationsausgabe verbaut.



Ich wollte gerne ein etwas größeres 1,3-Zoll-Display mit I2C-Anbindung einsetzen. Ein geeignetes OLED von Sainsmart hat den sh1106-Controller. Im Luma-Projekt gibt es Unterstützung dafür. Zuerst sollte man alle Abhängigkeiten installieren:

```
sudo apt-get install python-dev python-pip libfreetype6-dev libjpeg-dev build-essential
```

Und dann mittels *pip* die Python-Library:

```
$ sudo -H pip install --upgrade luma.oled
```

Über den Git-Hub geht das auch. Die Codebeispiele sind seit einiger Zeit noch an eine dritte Stelle ausgelagert worden, so dass sich auf jeden Fall noch

```
$ cd ~
```

```
$ git clone \
```

<https://github.com/rm-hull/luma.examples.git> (<https://github.com/rm-hull/luma.examples.git>)

```
$ cd luma.examples
```

```
$ sudo -H pip install -e
```

empfehlte und man diese Samples als Grundlage für die eigene Entwicklung auch noch parat hat. Das kann nicht schaden.

Ist das Display am I2C-Port des Pi richtig verbunden, am besten mit

```
$ i2cdetect -y 1
```

prüfen, ob an Adresse 0x3C das Display erscheint. Mit dem Python-Skript „demo.py“ aus den Samples kann man jetzt einen ersten Funktionstest wie folgt machen:

```
$ python demo.py [--i2c-port 1 --i2c-address 0x3C] -d sh1106
```

Wenn alles geklappt hat, sollte man eine solche Ausgabe bekommen:

**Display: sh1106**

**Interface: i2c**

**Dimensions: 128 x 64**

-----  
**Testing basic canvas graphics...**

**Testing contrast (dim/bright cycles)...**

**Testing display ON/OFF...**

**Testing clear display...**

**Testing screen updates...**

Unsere Zero Cam hat einen kleinen Kippschalter, der den Akku vom Pi und restlicher Elektronik trennt. Im ausgeschalteten Zustand bleibt sie durch die geringe Selbstentladung des LiPo lange betriebsbereit und man kann sie bis zum nächsten Einsatz im Schrank stehen lassen. Nach dem Einschalten wird das Startskript über einen *rc.local*-Eintrag mit

```
python zerocam.py -d sh1106
```

aufgerufen, wo alle notwendigen Bibliotheken eingebunden sind. Nach der Initialisierung der verbauten Hardware gibt es einen Statusscreen und nach Eingabe von Enter erscheint auf dem Display zuerst unser Hauptmenü:



**D – Shutdown****C – Upload****B – Löschen****A – Bild**

Für die Ablaufsteuerung erschien uns der Event-Handler des Touchphat gut geeignet, um bei Berührung eines der Panels entsprechend zu verzweigen:

```
def handle_touch(event):
    ... Bei jeder Eingabe ausführen ...

    if (event.name == 'A'):
        ... Bei Touch von Panel ,A‘ ...

    ...

    if (event.name == 'Enter'):
        ... Bei Touch von ,Enter‘ ...
```

Damit sind die vier wesentlichen Funktionen des Hauptmenüs abgebildet. Von hier aus lässt sich eine neue Aufnahme machen, kann man aufgenommene Bilder per FTP uploaden, den lokalen Bildspeicher auf der SD-Karte löschen und die Kamera herunterfahren.

Für die Flächen fünf und sechs des Touchphat (Enter, Back) sind auch schon passende Symbole aufgedruckt. Wird eine Aktion ausgeführt, gibt es entsprechende Informationen auf dem Display dazu. Damit man diese im Fehlerfall auch ordentlich lesen kann, kehrt man immer erst nach Back wieder zum Hauptmenü zurück.

Unterstützung gibt es auch durch die den Panels hinterlegte LED-Beleuchtung. So wird erfolgreiches Antippen durch kurzes Aufleuchten bestätigt, denn es fehlt ja die Haptik einer mechanischen Tastatur. Wird zur Quittierung ein OK oder ein Abbruch erwartet, dann lassen wir die in Frage kommenden Flächen ständig leuchten und sie erlöschen erst, nachdem man eine der geforderten Eingaben getätigt hat. Solcherlei Hilfe hat sich für den Außeneinsatz als recht günstig erwiesen.

Mit dem, was wir zum Bedienkonzept der Zero Cam entworfen haben, sind wir also recht zufrieden. Die Aufnahmequalität wird von den Kameramodulen für den Pi bestimmt. Unabhängig von der Version lohnt sich das kleine Festbrennweiten-Objektiv für Aufnahmen in die Landschaft bei gutem Licht. Für unsere Effektaufnahmen mit NoIR-Cam und Filtervorsatz braucht es viel Sonne und wenig Bewegung, dann werden sie gut.

Besonders glücklich bin ich, dass ich kein Provisorium mehr mit mir herumtragen muss, was aus lose zusammengefügte Teile besteht. Es ist eine kompakte Kamera geworden, bei der ich weiß, dass ich alles an Bord habe. Um festzustellen, ob ich vorher noch laden oder für neue Bilder Platz schaffen muss, reicht nach dem Einschalten nun ein Blick auf das Display...

-Anzeige-

**PC-WELT Marktplatz**

**PC-WELT Hacks - Technik zum Selbermachen?**

Raspberry Pi erfreut sich gerade unter Bastlern einer großen Beliebtheit. Kein Wunder, denn mit der 35-Platine lassen sich viele spannende Projekte realisieren. Vom Mediacenter, Netzwerkspeicher,

Fotomaschine bis hin zum Überwachungssystem ist alles möglich. Dieser Bereich ist aber nicht nur dem Raspberry Pi gewidmet, sondern bietet auch viele Tipps, Tricks und Anleitungen für andere spannende Bastelprojekte.

