



Version 2 ist da !

Liebe Unterstützer, Nutzer, Interessenten und Freunde,

auf diesen Newsletter habe ich mich wirklich gefreut. Ich darf euch allen ankündigen, dass die Software Version 2 verfügbar ist.

Die neuen Funktionalitäten begeistern mich und ich hoffe, dass ihr das alle auch so seht. Neben einer Generalüberholung aller Programme, die nun stabiler und vor allem robuster sein sollten, kann ich folgende neue Features ankündigen:

- es gibt ein automatisches Installationsskript !
- neue Konfigurationsseite mit vielen neuen Einstellmöglichkeiten
- Status-Website, die den CaravanPi vollständig untersucht
- Test Routinen für eine Reihe von Geräten am CaravanPi
- Speicherung der Messdaten in einer MariaDB Datenbank
- grafische Darstellung aller Messdaten über Grafana
- Versenden aller Messdaten an einen MQTT Broker (in der Cloud)
- Abfrage des Batteriemanagements einer LiPo Batterie (Liontron)
- Abfrage diverser Systemzustände des Raspberry Pi
- Abfrage und Überwachung der 230 V Versorgung und von 12 V Batterien
- Überwachung der Gaskonzentration im Caravan mit MQ-2 Sensor
- Testdaten Generator für die MariaDB
- Muster-Verarbeitung von Aktoren wie ESP und Shelly, mit denen Lichtschalter, Klimaanlagen, Heizungen gesteuert werden können

Ich freue mich schon sehr auf eure Reaktionen zu den Neuerungen. Lasst mich wissen, was ihr davon haltet.

Und natürlich würde ich gerne von euch wissen, wie es euch mit dem CaravanPi bisher ergangen hat. Ist der Einbau schon erledigt? Habt ihr die erste Fahrt mit den CaravanPi schon hinter euch? Was sind eure Erfahrungen? Habt ihr Wünsche oder weitere Ideen? Lasst mich teilhaben und schickt mir womöglich sogar Fotos eurer Installation. Nutzt dazu auch gerne den neuen (reaktivierten) [Discussions Bereich auf github](#).

Schon jetzt herzlichen Dank für euer Feedback!

Euer Sepp



Die Aktualisierung der Software dauerte leider eine ganze Weile - schon Ende 2022 hatte ich mich entschlossen, die Software zu überarbeiten. Immer wieder musste ich die Entwicklung für Monate unterbrechen, weil mich mein eigentlicher Job ziemlich in Atem hielt. Aber über Weihnachten kam ich nun gut voran - nicht zuletzt, weil ich Unterstützung durch eine KI hatte.

Die Software Entwicklung mit KI Unterstützung durch OpenAI's ChatGPT V4 begeistert mich. Die Geschwindigkeit, mit der ich vorankam, war beeindruckend und auf viele neue Ideen, die nun Teil der CaravanPi Software sind und zur Stabilität und Flexibilität beitragen, wäre ich alleine nicht gekommen. Zum Beispiel konnte ich das Framework FLASK vorher nicht. Nun ist es ein wesentlicher Bestandteil vom CaravanPi und löst u. a. das bottle-Framework und meine dann doch eher stümperhaft erstellten Konfigurationswebsites auf PHP Basis ab.

Ich kann jedem nur empfehlen sich mit den unterschiedlichen KIs zu beschäftigen. Natürlich muss man auch hierbei wissen, was man erreichen möchte. Aber auch Programmierunfahrene können so sehr schnell zu Ergebnissen kommen. Vielleicht sport euch das an, doch mitzuhelfen bei der Weiterentwicklung des CaravanPi.

Und git und github sind dank ChatGPT nun auch keine Bücher mit sieben Siegeln mehr für mich. Die KI half mir dabei, die Uploads sauber hochzuladen und ein neues Release zu veröffentlichen.

Die neue Website für Konfiguration und Bedienung

The screenshot shows the 'CaravanPi Grundeinstellungen' (Basic Settings) page. It includes fields for database storage (checkboxes for 'in Wertedateien speichern' and 'in der Datenbank speichern?'), and configuration for an MQTT Broker (checkbox for 'an MQTT Broker senden?' and fields for 'Adresse des MQTT Brokers', 'Port des MQTT Brokers', 'MQTT User', and 'MQTT Passwort').

CaravanPi abzufragen und einige Test-Routinen zu starten. Auch das Ansteuern vom ESP82xx oder Shellys ist über diese Oberfläche einfach möglich. Entsprechende Konnektoren vorausgesetzt können Sie nun die Beleuchtung oder die Klimatisierung über diese CaravanPi Website steuern.

Auch bisher hatte der CaravanPi schon eine Website um Konfigurationen vorzunehmen und Kalibrierungen zu starten. Die Site basierte auf HTTP und PHP und war nicht wirklich schön. Sie erfüllte ihren Zweck, das war aber auch schon alles. Mobil gut darstellbar war sie leider auch nicht.

Die neue Website für Konfigurationen und die Bedienung des CaravanPi basiert nun auf dem Framework Flask, das wiederum auf der Programmiersprache Python basiert. Da auch meine CaravanPi Software auf Python basiert, ist das nun eine ideale Kombination und ermöglicht viele neue Features.

Die neue Website bietet neben den bekannten Möglichkeiten die Konfigurationsdaten einzugeben auch die Möglichkeit den Status des

Konfigurationsdaten nun im XML Format und vielfältiger

Bisher waren die Konfigurationsdaten des CaravanPi in einzelnen Dateien gespeichert. Diese waren kaum lesbar und pflegen der Daten ohne die CaravanPi Konfigurationswebsite war schwierig.

Da ich in der Version 2 eine Vielzahl an neuen Konfigurationsmöglichkeiten vorsehen wollte, war das bisherige Format nicht mehr geeignet.

```
<CaravanPiConfigurations>
  <caravanpiDefaults>
    <countGasScales>1</countGasScales>
    <countTanks>1</countTanks>
    <write2file>1</write2file>
    <write2MariaDB>1</write2MariaDB>
    <send2MQTT>1</send2MQTT>
    <MariaDBhost>localhost</MariaDBhost>
    <MariaDBuser>caravanpi</MariaDBuser>
    <MariaDBpasswd>[REDACTED]</MariaDBpasswd>
    <MariaDBdatabase>CaravanPiValues</MariaDBdatabase>
    <MQTTbroker>[REDACTED]</MQTTbroker>
    <MQTPort>8883</MQTPort>
    <MQTTuser>CaravanPiMQTT</MQTTuser>
    <MQTTPassword>[REDACTED]</MQTTPassword>
    <countClimateSensors>1</countClimateSensors>
    <stromPInstalled>0</stromPinstalled>
    <intervalGasScale>10</intervalGasScale>
    <intervalTemp>10</intervalTemp>
    <intervalClimate>10</intervalClimate>
    <intervalTanks>10</intervalTanks>
    <intervalPositionNormal>10</intervalPositionNormal>
    <intervalRaspberry>10</intervalRaspberry>
    <LiontronMACAddress>[REDACTED]</LiontronMACAddress>
    <gasensorInstalled>1</gasensorInstalled>
    <gasensorDigitalIn>16</gasensorDigitalIn>
    <gasensorAnalogIn>1</gasensorAnalogIn>
    <gasensorAlarmActive>0</gasensorAlarmActive>
    <gasensorAlarmResume>0</gasensorAlarmResume>
    <v230CheckInstalled>1</v230CheckInstalled>
    <v230CheckADCPin>0</v230CheckADCPin>
```

Daher habe ich die Konfiguration auf eine neue Basis gestellt. Alle benötigten Daten werden nun in einer XML Datei gespeichert. XML ist ein universelles Format zum Verwalten von Daten innerhalb von Dateien. Das Format ist gut lesbar und kann auch manuell angepasst werden, falls die Konfigurationswebsite mal ausfallen sollte.

Die Konfigurationsmöglichkeiten haben sich dabei vervielfacht. Vieles was bisher fest im Programmcode verankert war, ist nun über die Konfigurationswebsite frei konfigurierbar. Auch werden über diesen Weg nun einige Funktionalitäten ein- bzw. ausgeschaltet werden.

Die Migration der bisherigen Konfigurationsdaten in das XML Format erfolgt dabei beim ersten Start des CaravanPis automatisch. Damit gehen ihre bisherigen Angaben zum Caravan etc. nicht verloren.

Statuswebsite zeigt den Zustand des CaravanPi an

The screenshot shows a web interface for monitoring the CaravanPi system. At the top, it says "CaravanPi". Below that, a message states: "Mit dem Aufruf dieser Website werden eine Reihe von Check-Routinen gestartet, die alle Komponenten des CaravanPi der Reihe nach durchtesten. Manche Tests benötigen **10 Sekunden und mehr**. Bitte haben Sie Geduld." A section titled "Prüfung der Komponenten" lists various components and their status:

Raspberry Pi	Raspberry PI 4B (b03115)
github CaravanPi Version	github.com: 2024-03-10 Raspberry: lokal hinter github: lokal vor github: 0 commits 0 commits
CaravanPi Defaults XML	OK
Python Bibliotheken	OK
MagicMirror	OK
MariaDB	Fehler
Grafana	OK
Apache Webserver	OK
1-Wire Bus	28-01144fea82aa 28-01144febdbaa 28-01144f936caa
I2C Bus	zu viele Devices! erwartet: 20,21,53,76 gefunden: 20,21,48,53,76 zusätzlich: 48
Wägezelle HX711	... Prüfung läuft ...
Netzwerk	
Crontab pi	
Cronstab root	

Apropos Protokolldatei: Auch hier gibt es eine Neuerung. Die Dateien können nun auch über die Website angesehen werden. Auf einen Blick ist die Größe der Log-Dateien sichtbar.

Und mit einem weiteren Klick kann man sich von einzelnen Dateien die letzten 30 Zeilen ansehen. Auch das hilft enorm bei der Fehlersuche.

Eine ähnliche Funktion ist die Anzeige des aktuellen Zustands des i2c Buses über die Website.

Auf der neuen Website gibt es auch eine Statusabfrage. Diese prüft nacheinander alle Komponenten des CaravanPi durch und gibt den entsprechenden Staus zurück. So kann die Fehlersuche schnell vonstatten gehen.

Die Statusabfrage nutzt dabei die in der Konfiguration hinterlegten Daten und passt die Prüfungen entsprechend an. Die einzelnen Prüfungsschritte sind meist schnell erledigt. Einige Schritte dauern aber auch eine kleine Weile, wie die Abfrage der Wägezelle.

Natürlich protokollieren die Prüfungsschritte ihre Ergebnisse in einer Protokolldatei.

position2file.log	0	2024-03-17 00:00:40
StromPi-RTC-sync.log	0	2024-03-17 00:00:40
systemstat.log	0	2024-03-17 00:00:40
tactileSwitches.log	0	2024-03-17 00:00:40
temp2file.log	549	2024-03-17 12:00:04
v12Check.log	0	2024-03-17 00:00:40
v230Check.log	0	2024-03-17 00:00:40

LOGs - jeweils letzte 30 Zeilen

- [backup.log](#)
► Details
- [bms2file.log](#)
► Details
- [climate2file76.log](#)
► Details
- [flask-main.log](#)
► Details

MariaDB / MQTT als neue Datenverarbeitungsmethoden

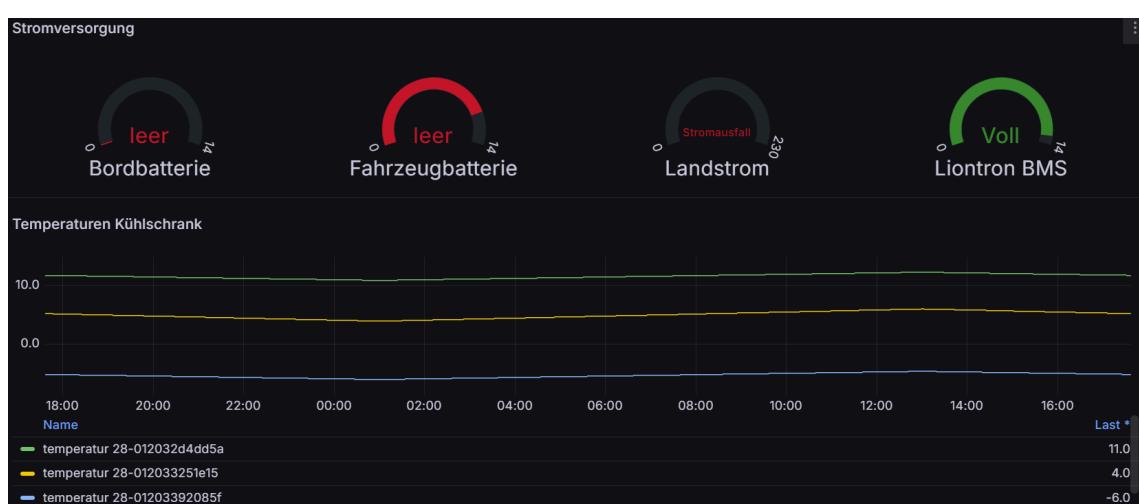
Bisher wurden alle Daten, die von den Sensoren eingelesen wurden, in einzelnen Werte-Dateien gespeichert. Das war bei Entwickeln der ersten Version die einfachste und schnellste Möglichkeit. Zudem konnte ich so die Verbindung zu den Modulen des MagicMirror herstellen. Die Nachteile nahm ich erst mal in Kauf. Aber die schwere Lesbarkeit und die Anfälligkeit dieser Methode ließen mir keine Ruhe.

Deshalb experimentierte ich mit einer MariaDB herum und mit ihr kam huckepack auch gleich Grafana daher. Doch der Reihe nach. Um eine SQL Datenbank und die Daten darin gut bearbeiten zu können, bietet sich phpMyAdmin an. Und mit den richtigen Python Modulen war die Implementierung in Kürze geschafft. Zusammen mit ChatGPT wurde daraus schnell eine universelle Methode für die Datenbank Operationen.

Jetzt sind die Daten der Sensoren in einer SQL Datenbank sicher gespeichert. Sie können dort in verschiedenster Weise ausgewertet werden. Das ist bei weitem besser als ASCII Dateien, die von logrotate in Zaum gehalten werden müssen. Und mit der richtigen Löschmethode gerät auch so nichts aus den Fugen.

Und da ich zu Hause meine Daten aus Hausautomation und anderen Sensoren mit MQTT zwischen meinen verschiedenen Servern und Installationen austausche, erweiterte ich diese Daten Schnittstelle auch noch um das Versenden an einen MQTT Broker. Damit können die Caravan Daten nun einfach z. B. an zu Hause geschickt werden. Aber auch das Versenden im "Netzwerk" des Caravans oder Wohnmobil ist kein Problem. Auch auf dieses Weise können Aktoren angesprochen werden. Doch dazu später mehr.

Grafana für die grafische Aufbereitung der Daten

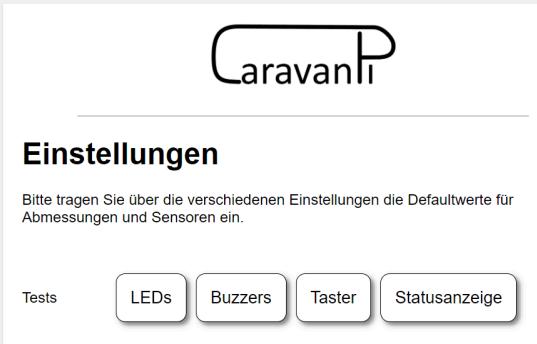


Grafana ist ein Tool, mit dem Daten aus Datenbanken und anderen Datenquellen grafisch aufbereitet und dargestellt werden können. Insbesondere Zeitreihen und Zustände können damit sehr einfach dargestellt werden. Grafana bietet dabei eine ganze Reihe von Möglichkeiten zur Umformung der Daten und deren Darstellung. Plugins erweitern diese Möglichkeiten noch mehr. Diese Grafana Panels können dann sehr einfach auf dem MagicMirror zur Anzeige gebracht werden.

Damit gewinnt der CaravanPi eine neue Dimension. Statt reiner Zustandsanzeige entsteht ein System, das Werteverläufe darstellen kann. Ein Temperaturabfall in der Nacht ist plötzlich sichtbar ebenso wie ein Spannungsabfall der Landstrom Versorgung oder der Bordbatterie. Aber auch reine Zustände wie "Landstrom da oder nicht", "Gasalarm ja oder nein" können grafisch ansprechend aufbereitet werden.

Insgesamt tatsächlich eine neue Dimension für den CaravanPi, auf die ich sehr stolz bin.

Testroutinen und Testdaten Generator



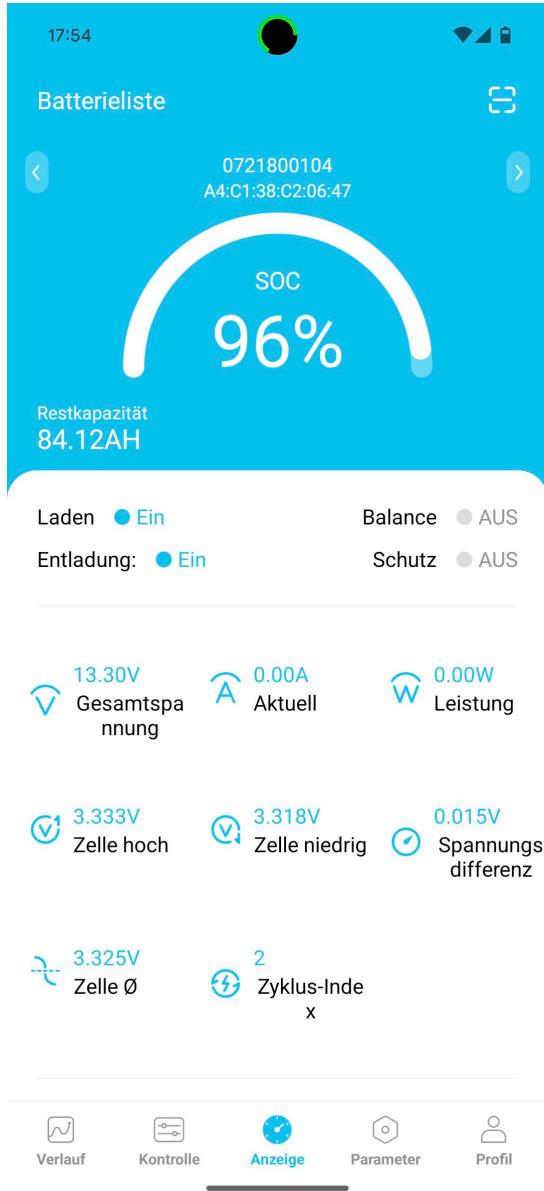
Auch die Tests für die LEDs, den Buzzer und die einzelnen Taster können nun sehr einfach aus der Website heraus gestartet werden.

Die Website ist wirklich ein echter Gewinn. Sie basiert auf dem Python Flask Framework und kann auch in Zukunft einfachst erweitert werden.

Da nun eine Datenbank verfügbar ist und die Anzeige auf dem Bildschirm vor allem über Grafana erfolgt, war es wichtig, relevante Testdaten in die Datenbank laden zu können, damit die Grafana Grafen auch ausprobiert werden konnten.

Ich habe Testdaten Generatoren erstellt, mit denen sich Testdaten erstellen lassen. Diese berücksichtigen "lebensechte" Datenverläufe und steuern die konkreten Zahlen per Zufall, so dass immer wieder neue Daten erzeugt werden können.

Ein Importskript schreibt die Testdaten dann in die Datenbank, wo sie dann für die grafische Auswertung durch Grafana zur Verfügung stehen.



Batteriemanagement einer LiPo Batterie abfragen

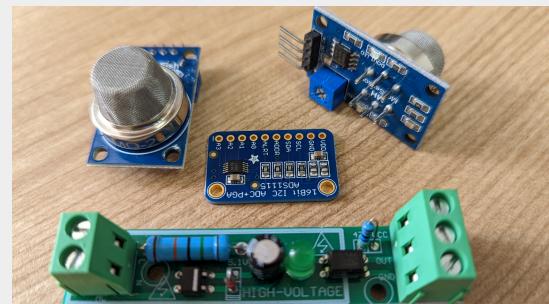
Ich habe kürzlich von einer Gelbatterie auf eine LiPo Batterie umgerüstet. Das beinahe halbierte Gewicht war dabei eines der Argumente. Aber auch das Batterie-Management dieser Batteriesorte interessierte mich sehr. Was kann man damit alles sehen?

Etwas enttäuscht war ich dann, also ich die Liontron App zum ersten mal mit einer Batterie verbunden hatte. Wirklich viele Informationen gab es darin nicht. Ich wusste aber, dass das Batteriemanagement in den Liontrons von XiaoXiang stammt. Und auch dieses Unternehmen biete eine [App](#) für Ihr Batteriemanagement, die allerdings bei weitem mehr Information aus der Batterie holt.

Es gibt eine ganze Reihe von Projekten, die versuchen, dieses Batteriemanagement über eine API abzufragen. Eines davon half mir, diese Abfrage in den CaravanPi zu integrieren. Damit ist ein erster Durchstich zu diesen Informationen geschafft. Die Zeit wird zeigen, welche Möglichkeiten sich damit noch bieten.

230V Landstrom, 12V Bord-/Fahrzeugbatterie, Gassensor

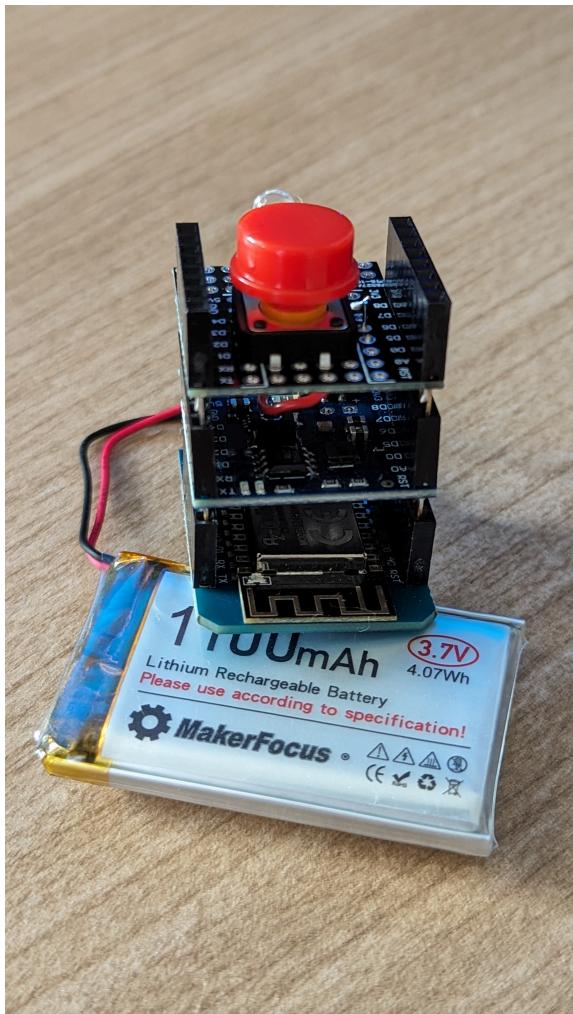
Bei unserem letzten Urlaub schaffte es unser Nachbar mehrmals hintereinander die Sicherung der Landstrom Versorgung zum Fliegen zu bringen. Dies war dann schließlich ein Problem für unseren Kühlschrank, der unsere Vorräte nicht mehr kühlen konnte. Gemerkt habe ich das nur durch Zufall. Daher wünschte ich mir eine Landstrom Überwachung für den CaravanPi.



Ich fand dafür ein kleines Breakout-Board, das ein analoges Signal liefert. Und da ich dann ohnehin einen Analog-Digital-Wandler benötigte, erdachte ich mir auch noch eine 12 Volt Überwachung für zwei 12 V Stränge (z. B. Bordbatterie und Fahrzeugbatterie).

Zudem war damit auch ein Gassensor für Propan/Butan möglich, den ich in dieses Analog Board auch noch integrierte. Über den Erweiterungsstecker der CaravanPi Platine für den I2C Bus war dieses Analog Board schnell in die Platine angebunden.

Aktoren einbinden ESP82 D1 mini, Shelly, ...



Ich werde immer wieder gefragt, ob der CaravanPi auch Klimaanlage, Heizung oder Küchengeräte ansteuern kann. Oder ob der CaravanPi die Ausrichtung des Caravans über entsprechende Motorstützen automatisieren kann. Meine Antwort war immer "im Prinzip, ja". Konkretere Hilfestellung konnte ich aber nicht bieten - bisher.

In der aktuellen Version habe ich die Ansteuerung von Aktoren über das flask framework beispielhaft integriert. Ich habe auf Basis eines ESP82 D1 mini einen Batterie betriebenen Taster realisiert. Diesen D1 mini habe ich zudem einer LED als beispielhaftes Empfangsgerät ausgestattet. Es hätte auch ein Relais sein können, mit dem Licht oder was auch immer gesteuert werden kann.

Ich konnte schließlich am CaravanPi eine Nachricht empfangen, wenn der Taster am batteriebetriebenen D1 mini gedrückt wurde. Auch das Loslassen des Tasters bekam ich mitgeteilt. Ausgehend von

diesem Signal steuerte ich dann vom Raspberry aus die LED an und brachte diese zum Leuchten. Das Aktivieren eines Relais ist genauso einfach möglich.

Ich konnte die LED aber auch einfach über die CaravanPi Website ein und ausschalten. Damit ist bewiesen, dass der CaravanPi beliebige Empfänger im Netz ansteuern kann und auch von beliebigen Aktoren angesteuert werden kann.

Der Erweiterung des CaravanPi mit Aktoren und beliebigen Empfängern steht also nichts im Weg.

Wie geht es weiter?

Platine V6

Beim Untersuchen fehlerhafter Werte vom HX711 erkannte ich, dass der HX711 etwas stabiler funktioniert, wenn er mit 5V anstatt 3V3 versorgt wird. Diese Änderung ist in die V6 der Platine eingeflossen. Zudem wurde der Pin 1 des 40 poligen Verbindungssteckers zum Raspberry Pi entsprechend markiert. Die Dateien zum Bestellen der Platine sind bereits online. Ich werde aber mit einer Bestellung noch warten, da ich zusammen mit JLCPCB

versuche, auch gleich Wannenstecker und JST XA Stecker für LEDs und Sensoren ab Werk auflöten zu lassen. Es sieht sehr gut aus.

Zusatzplatinen für Analogsignale bzw. das Bedienboard (z B. Deckel)

Ich entwickle gerade Platinenlayouts für das Analog Board und für das Board, das die Taster, LEDs und den Buzzer auf dem Deckel ansteuert. Auch das wird bestellbar werden.

Konfigurationsdatei XML

Die Möglichkeiten, die xml bietet, sind in der Version 2 noch nicht ausgeschöpft. Mit weiteren Konfigurationsdaten können die Programme des CaravanPi vom Nutzer noch einfacher an die eigenen Bedürfnisse angepasst werden. Hier habe ich noch eine Reihe von Ideen.

Sie sehen also, es tut sich auch weiterhin etwas beim CaravanPi. Ich bin Ihnen allen dankbar, wenn Sie mir Fehler oder gewünschte Verbesserungen melden. Ich versuche diese soweit möglich umzusetzen. Nutzen Sie dafür auch gerne die [Issues Funktion auf github](#). Dann bleibt alles gut verfolgbar.

Meine beschränkte Zeit ist und bleibt vorerst die größte Hürde für den weiteren Ausbau. Aber kommt Zeit kommt neue Version ...

Ihr
Josef Spitzlberger

CaravanPi - Josef Spitzlberger
Schloss Lustheim 1
85764 Oberschleißheim
josef.spitzlberger@caravanpi.de
<https://www.caravanpi.de>

[Datenschutz](#)
[Impressum](#)
[Newsletter abbestellen](#)

CaravanPi - Das Projekt
[github](#)
[Website](#)
[Wiki](#)
[Platine bestellen](#)