



HoneyPi

Hallo Bienenfreunde,

gerne stelle ich Euch hier unsere kleine Platine (HAT) für den **Raspberry Pi Zero W*** vor. Diese Platine soll als Basis dienen, die Verdrahtung sowie die Widerstände, Klemmsteine und/oder Pins aufzunehmen. Sie basiert auf den Wünschen vieler User, die sich in der Facebook-Gruppe oder auf <http://www.Honeypi.de> geäußert haben. Dabei sollen möglichst viele Sensorarten unterstützt werden, so wie es die HoneyPi Software (Version >/ =1.0) heute kann. Natürlich müssen nicht alle Bauteile oder Sensoren aufgebaut werden und auf Teile, die so klein sind, dass ich sie nicht erkennen kann (SMD 😊), habe ich hierbei verzichtet. Damit die Ausgänge des Raspberry nicht belastet werden, können Spannungswandler gesetzt werden.

Zum Aufbau musst Du also nicht Elektroniker sein, jedoch solltest Du mit dem Lötkolben und dem Multimeter umgehen können. Die Platine soll sowohl für einen einzelnen Bienenstock als auch für bis zu 5 Bienenstöcke einsetzbar sein. Dabei ist auch an eine nach und nach Erweiterung gedacht worden.

Hinweis in gemeinsamer Sache:

Ich bin weder Elektroniker, noch Ingenieur für Elektrotechnik, o. ä. fachlich ausgebildet und entsprechend weise ich darauf hin, dass jeder dieses Projekt eigenverantwortlich für sich nutzen kann. Die eingesetzten Teile habe ich für mich zum Testen gekauft und habe in diesem Zusammenhang keine Zuwendung erhalten. Herstellerbezeichnungen, Links zu Händlern, o.ä. dienen nur der Referenz. Kauf Deine Teile, wo Du diese am besten und günstigsten bekommen kannst. Eine strukturierte Einkaufsliste ist im Unterkapitel „Teilebeschaffung“ verlinkt.

Das eine oder andere an Infos habe ich aus dem Internet, wobei ich nicht mehr genau weiß woher. Neben <http://www.Honeypi.de> war das Forum von www.mikrocontroller.net eine sehr gute Quelle. Gute Grundlagen sind auch unter <https://www.elektronik-kompendium.de> zu finden.

Eine monetäre Nutzung, also den Verkauf der Platine mit dem Ziel Geld zu verdienen, ohne meine Genehmigung, untersage ich. Die Nutzung, auch als Basis für eigene, nicht kommerzielle Projekte und ein Hinweis auf meine Arbeit, würde mich sehr freuen.

*Gerne erstelle ich, mit Dir zusammen, später auch ein Layout für den Raspberry Pi 3 oder 4. Wenn ein WittyPi mini eingesetzt wird, kann auch ein Raspberry Pi 3 oder 4 eingesetzt werden.

Mitgewirkt haben: Javan R., Christian R., Christian W., Ulrich R.



Inhalt

Platine v2.xx.....	5
1. Funktionsliste des Boards:.....	5
• Spannungsversorgung und Messung	5
• Taster (Wartungsmodus = an/aus)	5
• LED	5
• Gewichtsmessung	6
• Temperaturmessung 1-Wire	6
• Diverse I ² C Bus-Sensoren	6
• WittyPi.....	6
• Timer Board.....	6
2. Spannungsversorgung.....	7
• Übersicht	7
• LM2596 Modul (Einspeisung 12 V)	7
• XL4015 Modul (Einspeisung 12 V)	8
• Einspeisung 5 V.....	8
• Versorgung der Sensormodule mit Festspannungsregler LF33CV für 3,3 V.....	8
• Muss ich einen Spannungsregler verwenden?	9
• Kann ich auch einen anderen Spannungsregler als den LF33CV verwenden?	9
• Ministückliste LF33CV	10
• Brücken (Jumper) HX711.....	10
• Brücken (Jumper) I ² C.....	10
• Brücken (Jumper) DS18B20 (Spannungsversorgung)	11
3. Absicherung	12
4. Spannungsmessung.....	13
• PCF8591 (I ² C)	13
• Spannungsteiler.....	14
• Musterberechnungen: Spannungsteiler	14
• Ministückliste PCF8591	15
5. Taster.....	15
• Taster entprellen.....	15
• Musterberechnungen: Grenzfrequenz f ₀	15
• Ministückliste Taster	15
6. LED.....	16
• Vorwiderstand für die LED	16
• Ministückliste LED	16
7. Gehäuse	17
• Übersicht	17
• Höhenprofile Raspberry Pi-HAT-LM2596 (XL4015)	17
• Kabel & Stecker.....	18
• Kabel für die Spannungsversorgung	18
• Datenleitungen	18
8. Raspberry Pi Anschluss (40 pol.).....	20
• Ministückliste Stacking Header	20



HoneyPi

9.	Gewichtsmessung	20
	• HX711 A/D-Wandler für die Wägezellen.....	20
	• Anschluss, Kabel und Abschirmung	21
	• HX711 Modulübersicht.....	21
	• Typ „Grün“ analog GND(E-) und digital GND verbinden.	22
	• Typ „Grün“ Sampling-Rate erhöhen	22
	• Typ „Violett“	22
	• Typ „Red Sparkfun“	22
	• Typ „Red“	22
	• Typ „Grove“	22
	• Typ „XFW“.....	22
	• Wägezelle	22
	• Wägezelle und EMV-Schutz	23
	• Temperaturdrift.....	23
10.	Temperatursensor DS18B20 (1Wire)	23
	• Ministückliste DS18B20	24
11.	I ² C Schnittstellen / Sensoren.....	24
	• Übersicht	24
	• BME680	24
	• BME280	25
	• 4 Pin Module / Lötbrücke BME280	25
	• Ministückliste BME280/BME680.....	25
	• Alternative I ² C Sensoren	26
12.	SHT20 (nicht kompatibel mit SHT3x / noch nicht vollständig umgesetzt)	26
13.	I ² C Adressenübersicht	27
	• I ² C Adresserweiterung	27
14.	Mehr GPIO Ports (CJMCU-2317) für die Menüsteuerung des OLED	28
15.	Kleinteile.....	28
16.	Sensor Board HX711 und DS18B20	28
17.	Kabelverschraubung Tabelle.....	31
	• PG Kabelverschraubung Tabelle (PG-Durchmesser Tabelle)	31
	• Metrische Kabelverschraubung Tabelle	31
	• Muster Bohrbild	31
18.	Software Konfiguration.....	31
19.	Tabelle Verbraucher	31
20.	Beispiel für eine Stromüberschlagsrechnung.....	32
	• Summe des 1. Stromkreises (3,3 V):.....	32
	• Summe des 2. Stromkreises (5 V):.....	32
21.	Belegungstabelle 40pol. / GPIO / PIN / Device.....	34
22.	Belegungstabelle GPIO	35
23.	Teilebeschaffung und Einkaufsliste	35
24.	Ausblick.....	36
25.	Bauanleitung(en)	37
	• Vorbereitung oder „Wie fange ich an“?	37



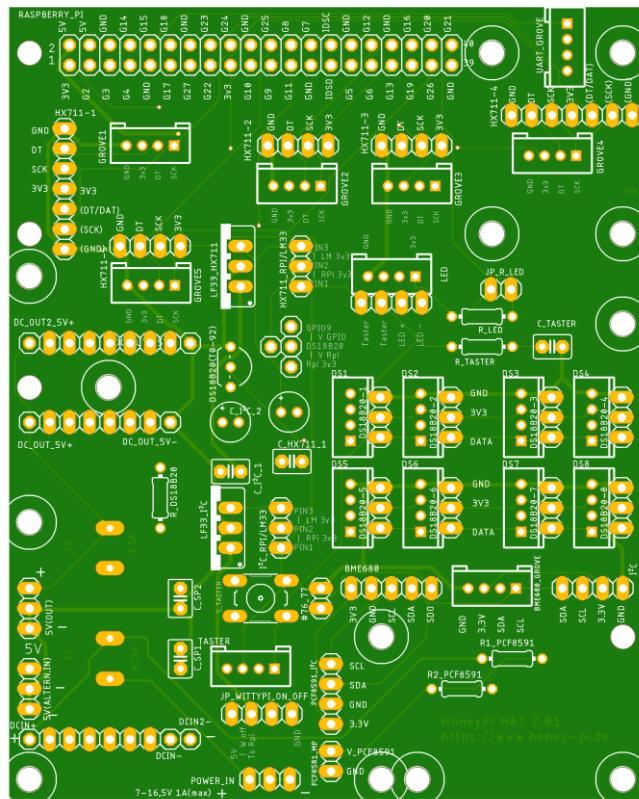
HoneyPi

• Bauvorschlag I.....	37
• Materialliste und Beschaffung:	37
• Aufbau:.....	37
26. Trouble-shooting guide / FAQ / Tipps.....	42
• Keine Spannung?	42
• Raspberry bootet nicht von derr SD-Karte.....	42
• Taster geht nicht.....	42
• Sensoren gehen nicht / bei Steckverbindern.	42
• HX711	42
• Uhrzeit.....	42
• Generelles zum I ² C Bus (Links):.....	43
• I ² C Bus Teilnehmer prüfen:	43
• I ² C Bus Verbindungsausfälle, o.ä. Probleme:	44
• I ² C Bus: Pullups reduzieren	44
• I ² C Bus Stromverbrauch und Pullups:	44
• I ² C Bus: Übertragungsfrequenz reduzieren & Reichweite erhöhen	45
• Geht nicht ;-(...)	45
27. Änderungshistorie Platine	46
28. Änderungshistorie des Dokuments.....	47



HoneyPi

Platine v2.xx



Die Platine 2.01 (Maße 100 x 80 mm, 2 Layer) habe ich mit Eagle CAD erstellt und bereits einmal (5 St.) bestellt und getestet. Die notwendigen Dateien (Gerber Files) zur Bestellung bei einem Fertiger (und auch die Quelldateien) stehen [bei GitHub](#) in aktueller Version bereit. Meine Platinen bestelle ich bei <https://jlpcb.com/>.

1. Funktionsliste des Boards:

 Spannungsversorgung und Messung

LM2596

XL4015

5 V extern

Versorgung Platine (der Sensoren) vom Raspberry PI

(Micro USB / USB-C → Pin-Header)

Spannungssensor Schnittstelle

WittyPi Schnittstelle

5 V Schnittstelle für weitere Geräte

2x LF33CV zur Erzeugung einer 3,3 V Spannung aus der 5 V Spannung

Taster (Wartungsmodus = an/aus)

OnBoard Taster (Printmontage) und Steckverbinder (1x RM2.0 mm für Grove, 1x RM2,54 mm für Pins, Klemmsteine o.ä.) für einen Gehäusetaster.

 LED

(blinkt ~3x, wenn Bootvorgang abgeschlossen und leuchtet im Wartungsmodus)

1x RM2.0 mm für Grove



HoneyPi

1x RM2,54 mm für Pins, Klemmsteine, direktes Auflöten

Vorwiderstand ein/ausschaltbar über Jumper für LEDs, die mit einem Vorwiderstand ausgerüstet sind.

Gewichtsmessung

3+2x HX711 (2x könnten aufs Board gelötet/gesteckt werden, wobei das ausdrücklich nicht empfohlen wird). Auch muss eine Abstützung der kl. Patine erfolgen.

Aufgelötet werden können 2x grüne oder 2x violette Module

Es ist besser, die fünf HX711 Schnittstellen zu den Sensor Boards (siehe Sensorboard 2.x) zu verlegen und auf den Sensor Boards ein HX711 Modul zu nutzen.

Temperaturmessung 1-Wire

Anschlussmöglichkeit für 8x RM 2.0 mm für Grove oder RM 2,54 mm für Pins, Klemmsteine o.ä.) auch als Schnittstelle zum Sensor Board 2.x

1x DS18B20 (Bauform: TO-92)

→ Zukünftig werden die 1-Wire Anschlussmöglichkeiten reduziert. Sie werden dann auf dem Sensorboard zu finden sein.

Diverse I²C Bus-Sensoren

z. B. BME680, BME280, PCF8591

Schnittstelle zum I²C Sensor Board 2.x, PCF8591, BME680(BME280)

Spannungsversorgung (LF33CV, 2x Kondensator, Jumper)

WittyPi

Hierfür sind Pins vorhanden, um die Stromversorgung über das WittyPi Board umzuleiten und somit das Ein-/Ausschalten des Raspberry Pi, durch das WittyPi Board zu ermöglichen.

Timer Board

Gedacht als einfache Lösung anstelle des WittyPi.

Die Hardware funktioniert. Die softwareseitige Unterstützung durch HoneyPi wurde noch nicht angegangen.



HoneyPi

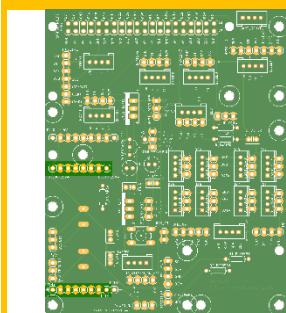
2. Spannungsversorgung

Auf der Platine sind 2x 3 Pin-Holes im RM 5,08/2,54mm für die Spannungseinspeisung vorgesehen. Die eine Variante dient zur Einspeisung von 5 V und die 2. Variante für größere Spannungen, die über aufsetzbare DC-DC Konverter in 5 V gewandelt werden können.

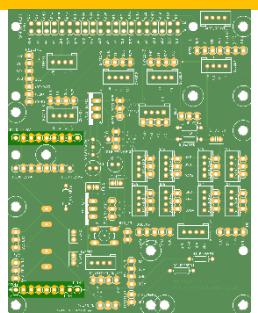
Es können hier die größeren RM 5,08 mm Klemmsteine und alternativ auch die kleineren 2,54 mm Klemmsteine eingesetzt werden. Dazu sind von den 3 Pin-Holes ein Pin-Hole für + und 2 für GND vorgesehen. Der kleine Klemmstein kommt in die Pin-Holes 1 und 2 und der große Klemmstein in die Pin-Holes 1 und 3.

⚠ Übersicht

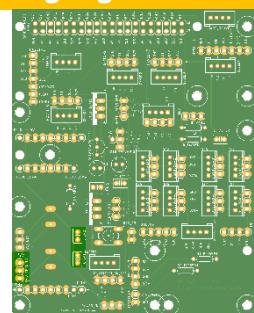
Anschluss LM2596



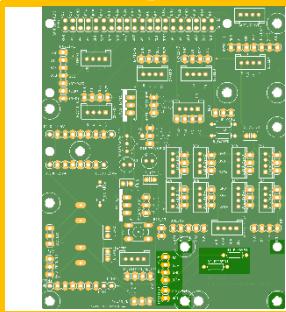
Anschluss XL4015



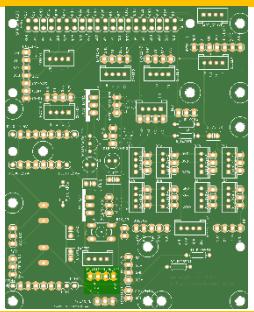
Eingang 5 V



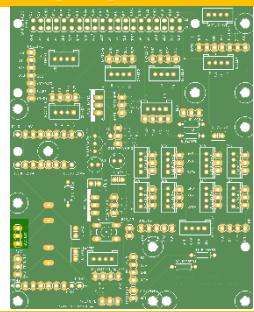
Spannungssensor



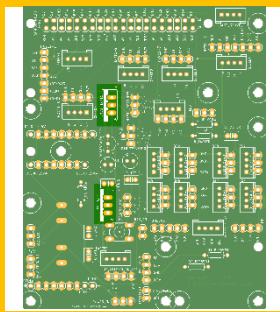
WittiPi



Ausgang 5 V

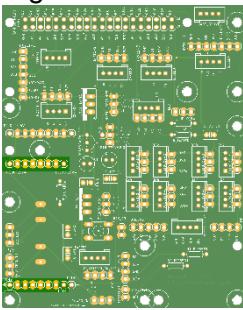


LF33CV



⚠ LM2596 Modul (Einspeisung 12 V)

LM2596 Modul kann z. B. bei Solaranlagen die Spannung von ~12 V auf die benötigten 5 V regeln.



⚠ Beim Aufbau ist der Hinweis in-/out- (roter Kreis) zu beachten und die Ausgangsspannung von 5 V muss vorab am Potentiometer (Foto, blaues Bauteil) und einem Voltmeter an +/- out eingestellt werden!



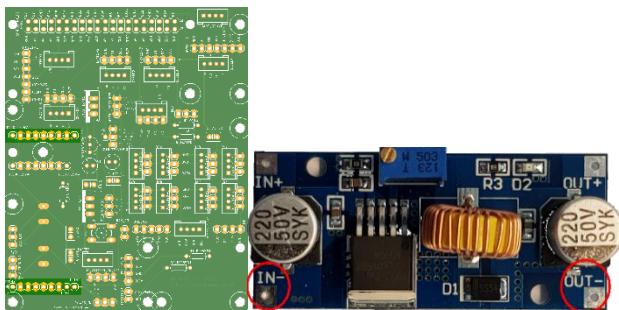
HoneyPi

⚠️ Bei einer Verpolung kann der LM2596S bzw. einer der Kondensatoren auf dem Modul platzen. Die Keramik des LM2596 zerspringt dabei und es können Körperschäden auftreten!

💡 XL4015 Modul (Einspeisung 12 V)

Alternativ z.B. für eine Solarversorgung.

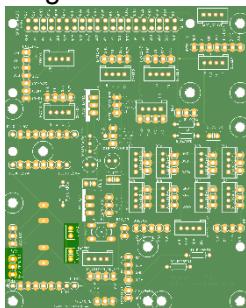
Im Heise.de Forum habe ich von „spontanen“ Ausfällen des LM2596 gelesen. Daher habe ich dieses alternative Modul vorgesehen. Auch wenn weitere Verbraucher geplant werden sollten, ist dieses Modul eine Alternative, da ein wesentlich höherer Strom bereitgestellt werden kann.



- ⚠️ Beim Aufbau ist der Hinweis in-/out- (roter Kreis) zu beachten und die Ausgangsspannung von 5 V muss vorab, ohne den Anschluss weitere Bauteile, am Potentiometer (Foto, blaues Bauteil) und einem Voltmeter an +/- out eingestellt werden!**
- ⚠️ Bei einer Verpolung kann der XL4015 bzw. einer der Kondensatoren auf dem Modul platzen.**

💡 Einspeisung 5 V

Pin-Holes 5,08 / 2,54 mm z.B. für Solarversorgung mit Solarregler der eine „saubere“ 5 V Spannung bietet. Um bei Bedarf die Spannung zu glätten, können zwei Kondensatoren (z. B. ~1 µF und 100 µF) aufgelötet werden. Gezeichnet wurden Keramikkondensatoren (Kerko). Die eigene Auslegung zur Spannungsversorgung kann auch einen Elektrolytkondensator (Puffer) ergeben. Zur Erleichterung habe ich ab der Version 2.02 ein kl. + Zeichen eingezeichnet.



💡 Versorgung der Sensormodule mit Festspannungsregler LF33CV für 3,3 V

Für die 3,3 V Spannungsversorgung der HX711 Module und der Module am I²C Bus haben wir eine konfigurierbare Möglichkeit geschaffen, dass die Spannung nicht über den Raspberry bereitgestellt wird, sondern über je einen Festspannungsregler Typ LF33CV. Ohne Elko (Elektrolytkondensator) schwingt die Spannung um etwa +/- 0,2 V mit einem Elko um 22 µF wird das Schwingen auf ca. +/- 0,01 V reduziert. Das Datenblatt gibt eine mind. Kapazität von 2,2 µF vor und in Musterapplikationen wird ein 10 µF Elektrolytkondensator eingesetzt. Das Verhalten des 10µF Elkos konnte ich nicht mehr visualisieren, da ich keinen



HoneyPi

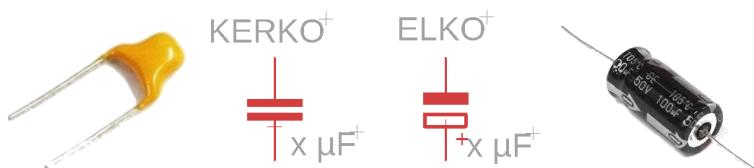
Zugriff auf ein Oszilloskop mehr habe. 10 µF und 22 µF sind in meinen beiden Aufbauten verbaut und laufen unauffällig/gut.

Wenn Du eh bestellen musst, rate ich zu einem 10 µF Elektrolytkondensator. Hast du etwas zwischen 2,2 µF und 22µF vorrätig, dann probiere dies aus.

- Wesentlich grösser macht keinen Sinn, bzw. kann sogar die Regelung des LF33CV verlangsamen/verschlechtern.

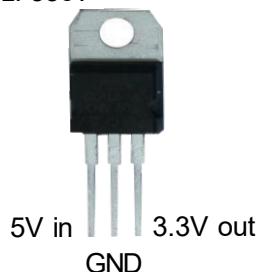
⚠ Elektrolytkondensatoren dürfen nicht falsch (verpolt) eingebaut werden. Meist ist der Minuspol gekennzeichnet.

Typisches Aussehen von Keramik-(ii) oder Elektrolytkondensator(re):



Eingangsseitig ist ein 0,1 µF/100 nF Keramikkondensator (Kerko) vorgesehen. Dieser dient der Vermeidung von hochfrequenter Oszillation. Ein Kerko hat so gut wie keine parasitäre Induktivität.

LF33cv



ⓘ Muss ich einen Spannungsregler verwenden?

Nein! Es kommt drauf an, was Du alles anschließen möchtest.

Ich belaste meinen GPIOs nur mit 10mA und alle zusammen mit max. 30 mA.

ⓘ Kann ich auch einen anderen Spannungsregler als den LF33CV verwenden?

Klar, den musst Du dann allerdings selber auslegen. Neben der Pinbelegung können auch andere Eigenschaften anders sein.

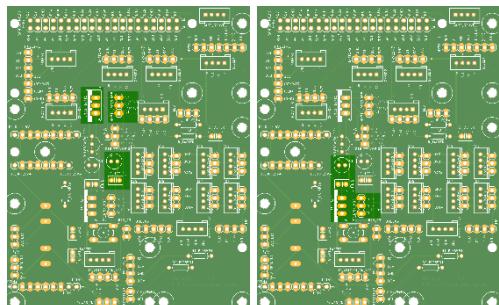
Ein Beispiel: Fertige Breakoutboards enthalten oft den AMS1117. Er könnte oberflächlich gut passen, benötigt aber mindestens 10mA Last. Kommt man nicht in den Bereich >10mA, riskiert man, dass der Spannungsregler aufgrund einer zu geringen Last eine zu hohe Spannung ausgibt, bzw. schwingt.

Der Spannungsregler LF33CV fällt nicht, wie ggf. andere Spannungsregler gleich aus, wenn die Eingangsspannung zu gering ist, sondern er liefert dann eine entsprechend geringere



HoneyPi

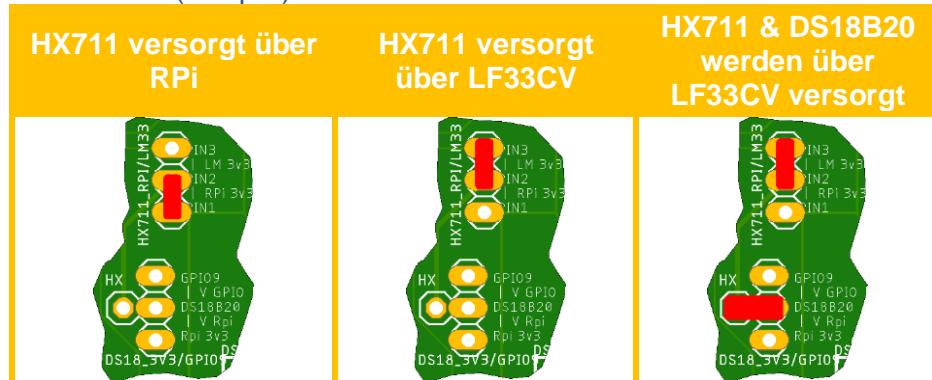
Ausgangsspannung. Beispielsweise sind bei 3,1 V Batteriespannung, ungefähr 2,8 V Ausgangsspannung zu erwarten.



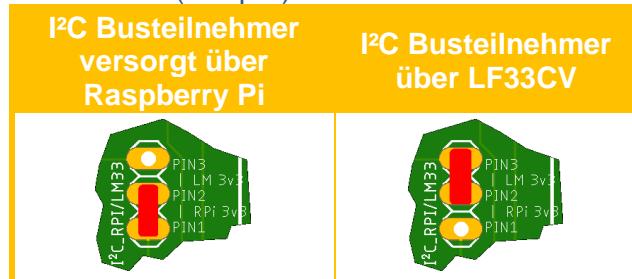
Ministückliste LF33CV

Menge	Bezeichnung	LINK
1-2	LF33CV	Aliexpress
1-2	Keramikkondensator 0,1 µF/100 nF, >/ = 5 V	
1-2	Elektrolytkondensator 10 µF, >/ = 5 V	

Brücken (Jumper) HX711



Brücken (Jumper) I²C





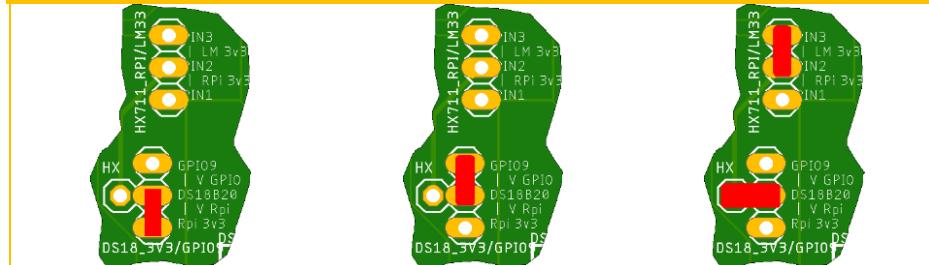
HoneyPi

Brücken (Jumper) DS18B20 (Spannungsversorgung)

DS18B20 versorgt
über Raspberry Pi
(3,3 V)

DS18B20 versorgt
über Raspberry Pi
(GPIO09)

DS18B20 versorgt
über Raspberry Pi





HoneyPi

3. Absicherung

Das Du Deine Anlage eigenverantwortlich aufbaust, ist klar. In dem Zusammenhang musst Du Dich mit dem Thema „Absicherung“ beschäftigen und informieren. Du denkst jetzt vielleicht „meine Photovoltaikanlage hat doch nur 18 V und die Batterie nur 12 V“, daher ein kurzes Beispiel aus meiner „Jugend“. Bei einer LKW-Reparatur ist mir ein 17er Maulschlüssel auf den Anlasser-Pluspol gefallen und hat dann den Kontakt zur Karosseriemasse hergestellt (Ich wollte nur mal schnell...). Die Leitung war nicht (und sie ist es standardmäßig auch heute nicht) abgesichert und direkt mit der Batterie verbunden. Der Stromfluss hat den Maulschlüssel zum Glühen gebracht und das ganze Fahrzeug hätte abbrennen können! Ich hatte Glück und deswegen solltest **DU** Dir Gedanken zu einer Absicherung Deiner Anlage machen!

Leitungssicherung (Hauselektrik)	Leitungssicherung (KFZ-Bereich)
	

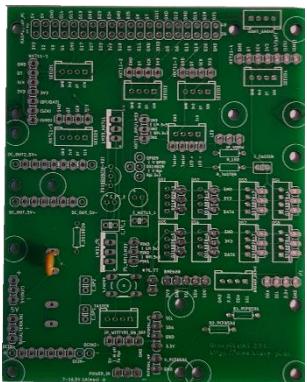
⚠ Sicherungen Lösen beim ~1,5 - 1,6-fachen des Nennstromes aus. Genaue Daten findest Du in den Datenblättern.

Zur Absicherung der Platine sind zwei Möglichkeiten zum Anschluss einer Sicherung vorhanden.

1x (F1) für die Eingangsspannung vor den Spannungsreglern (LM2596 o. XM4015) und 1x (F2) bei 5 V Spannungsversorgung (nach dem Spannungsanschluss).



HoneyPi

Sicherung F1 vor dem Spannungswandler bei Einspeisung von Spannungen >5 V	Sicherung F2 bei 5 V Einspeisung (ohne Spannungswandler)
	

Ich nutze, mit dem Wissen, dass diese sehr träge sind, Thermosicherungen, auch als Polyfuse, Multifuse, Polyswitch benannt.

Wenn Du außerhalb der Platine eine Sicherung vorsehen möchtest, z.B. weil Du eine Schmelzsicherung bevorzugst, dann brücke hier mit einem Draht.

Ich setze, **vor** dem LM2596, eine 0,25A Polyfuse (F1) ein. Beim Einsatz eines UMTS Modems wird diese ggf. nicht reichen. Hier würde ich max. 1A ansetzen.

Bei der Verwendung des 5 V Anschlusses, also ohne Spannungswandler, kann die 2. Position (F2), mit einer Sicherung für eine zu bestimmende Stromstärke, genutzt werden.

4. Spannungsmessung

PCF8591 (I²C)

Der PCF8591 kann über 3 Stehbolzen (M2,5, z. B: Lg.6mm) auf das Board gesetzt werden und über Kabel angeschlossen werden. Bei gewinkelten Pins, sollte zumindest eine Pin-Reihe umgebogen werden (s. Bild).

Der PCF8591 kann theoretisch bis zu 6 V am Messkanal aushalten aber nur max. bis zur Versorgungsspannung messen. Wir haben uns für eine Versorgungsspannung von 3,3 V für alle I²C Teilnehmer entschieden. D.h. die max. messbare Spannung ist 3,3 V. Dazu können die beiden Widerstände für einen Spannungsteiler mit auf die Platine gelötet werden. Z.B. 30 kΩ + 7,5 kΩ (oder deren Vielfaches). Der Anschluss zur Spannungsmessung heißt V_PCF8591 und sollte mit AIN2 auf dem PCF8591-Modul verbunden werden.

Bei der Verwendung von AIN0,1,3 sollten alle Jumper entfernt werden.





HoneyPi

⚡ Spannungsteiler

Das Verhältnis 4:1 sollte für eine 12 V Anlage ($V_{MAX} = 16,5$) sein.

Um den permanenten Stromverbrauch so gering wie möglich zu halten, sollte $R_{Ges.}$ möglichst groß sein. Gute Erfahrungen haben wir mit 30 kΩ / 7,5 kΩ oder 40 kΩ / 10 kΩ gemacht.

In dieser Konstellation werden durch den Spannungsteiler 16,5 V als 3,3 V auf den Messkanal gegeben.

Bei anderen Spannungen können/müssen andere Widerstände eingesetzt und eine Anpassung in der HoneyPi Oberfläche gemacht werden.

Der Gesamtwiderstand $R_{Ges.}$ darf dabei nicht zu niedrig und nicht zu hoch sein.

⚡ Musterberechnungen: Spannungsteiler

12 V, 30 kΩ / 7,5 kΩ	12 V, 40 kΩ / 10 kΩ
$V_{max} = 16,5 \text{ V}$	$V_{max} = 16,5 \text{ V}$
$R_1 = 30.000 \Omega$	$R_1 = 40.000 \Omega$
$R_2 = 7.500 \Omega$	$R_2 = 10.000 \Omega$
$R_{Ges.} = 37.500 \Omega$	$R_{Ges.} = 50.000 \Omega$
$P = U * I$ und $I = U / R$	$P = U * I$ und $I = U / R$
$P = U^2 / R$	$P = U^2 / R$
$P = 16,5 \text{ V} * 16,5 \text{ V} / 37500 \Omega$	$P = 16,5 \text{ V} * 16,5 \text{ V} / 50.000 \Omega$
$P = 0,00726 \text{ W}$	$P = 0,005445 \text{ W}$
$I = P/U$	$I = P/U$
$I = 0,00726 \text{ W} / 16,5 \text{ V}$	$I = 0,005445 \text{ W} / 16,5 \text{ V}$
$I = 0,00044 \text{ A}$	$I = 0,00033 \text{ A}$
$I = 0,44 \text{ mA}$	$I = 0,33 \text{ mA}$



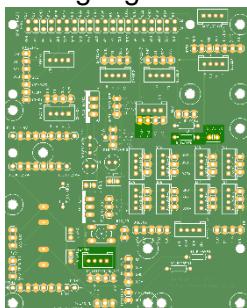
HoneyPi

Ministückliste PCF8591

Menge	Bezeichnung	LINK
1	PCF8591	Aliexpress
1	Widerstand 30 kΩ (≤1%)	
1	Widerstand 7,5 kΩ (≤1%)	
1 Set	Stehbolzen und Schrauben	Aliexpress

5. Taster

Neben der Möglichkeit einen Kurzhubtaster direkt auf die Platine zu setzen, kann natürlich auch ein Taster für das Gehäuse eingesetzt werden. Dafür stehen Pin-Holes für Grove Stecker, als auch die RM 2,54 mm Pin-Holes z.B. für Klemmsteine, Dupont-Stecker zur Verfügung. OnBoard parallel zum Gehäuseeinbautaster ist möglich.



Taster mit integrierten LEDs können einen Vorwiderstand haben. Dieser sollte für 3,3 V passen. Leider sind bei vielen angebotenen LED-Tastern keine ausreichenden Daten angegeben. Daher halte ich es für besser, LED und Taster getrennt zu kaufen.

Taster entprellen

Das eigentliche Entprellen wird per Software ausgeführt. Des Weiteren können ein Widerstand und ein Kondensator aufgelötet werden. Diese Kombination nennt man passiver Tiefpassfilter 1. Ordnung.
Dabei soll der Tiefpassfilter die hohen Frequenzen abschwächen, beziehungsweise sperren und niedrige Frequenzen durchlassen.

Musterberechnungen: Grenzfrequenz f_0

Muster 1

$$f_0 = 1/(2\pi R C)$$

$$f_0 = 1 / (2\pi \cdot 50.000 \Omega \cdot 0.000.000.1 f)$$

$$R = 47 \text{ k}\Omega$$

$$C = 100 \text{ nF}$$

$$f_0 = 33.86 \text{ Hz}$$

Default (OnBoard Taster)

$$R = 50 \text{ k}\Omega$$

$$C = 22 \text{ nF}$$

$$f_0 = 144,68 \text{ Hz}$$

Ministückliste Taster

Menge	Bezeichnung	LINK
1	Einbautaster	Amazon / Metzler



HoneyPi

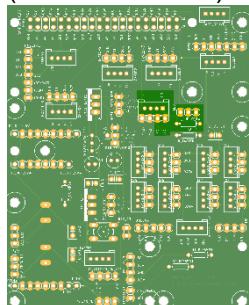
Menge	Bezeichnung	LINK
1	Taster für die Platine	Aliexpress
1	Kondensator, Keramik, 22nF, >= 5 V	
1	Widerstand 50 kΩ (≤1%)	

6. LED

Hier sind wir von einer LED ausgegangen, die im Gehäuse platziert wird.

Da es LEDs mit Vorwiderstand (z.B. auf kleinen Platinen) gibt, kannst Du den Vorwiderstand per Jumper oder Lötbrücke ein und ausschalten. Natürlich kannst Du den Vorwiderstand dann auch weglassen 😊

Bei der Grove-Buchse „LED“ sind nur Pin 1(LED+) und Pin 4 (LED- /GND) belegt. Der Taster wird weiter unten angeschlossen (siehe Taster). Die beiden unbelegten Pins sind mit NotC (Not Connected) beschriftet.



💡 Vorwiderstand für die LED

Der Vorwiderstand soll den Strom begrenzen, also verhindern, dass die „LED“ durchbrennt und dass ggf. nicht unnötig viel Strom verloren geht.

Eine rote 2V/10mA benötigt einen Vorwiderstand von 130 Ω. Damit Du nicht rechnen musst, habe ich eine Tabelle mit Standardwerten zur GPIO Spannung von 3,3 V erstellt:

Farbe	Spannung	Strom	Vorwiderstand*
Rot	2,0 V	10 mA	130 Ω
Gelb	2,1 V	10 mA	120 Ω
Grün	2,2 V	10 mA	110 Ω
Rot	1,7 V	2 mA	800 Ω
Blaugrün	3,2 V	20 mA	5 Ω
Rot	2,0 V	30 mA	43 Ω

*Meist kann man auch einen etwas größeren Widerstand nutzen. Dann leuchtet die LED nicht ganz so hell (bei einem zu hohen Widerstand gar nicht mehr). Der Stromverbrauch sinkt, was aber aufgrund der geringen Betriebszeit (nur Wartungsmodus) eher unwesentlich ist. Denk bitte dran, dass der Strom, der aus dem Raspberry Pi GPIOs fließt nicht zu hoch sein darf. Meine persönliche Grenze ist bei 30mA, wissend, das vielleicht etwas mehr geht. Bei einer Betrachtung sind alle 3,3 V Verbraucher am Raspberry Pi, die zur gleichen Zeit aktiv sind, zu beachten!

💡 Ministückliste LED

Menge	Bezeichnung	LINK
-------	-------------	------



HoneyPi

1	LED 2,1V / 12 mA Schutzart IP 67	Reichelt / Signal Construct SMCP 06
1	100 Ω Vorwiderstand	

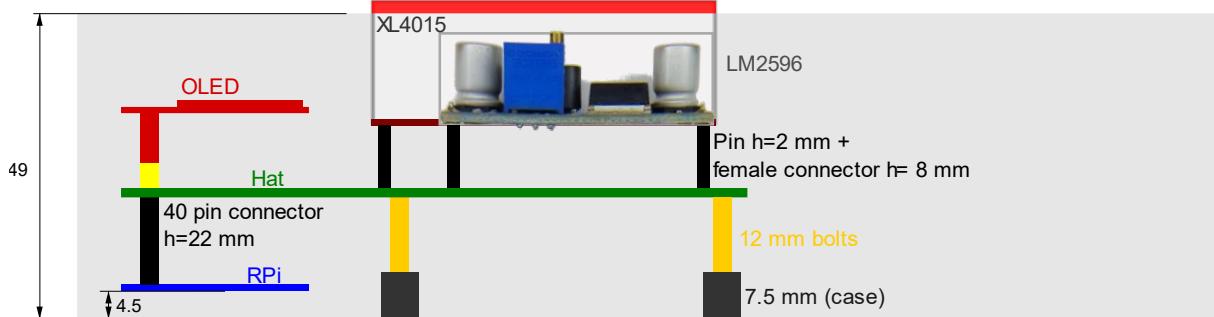
7. Gehäuse

Als Gehäuse habe ich mich für ein einfaches Kunststoffgehäuse 190 x 190 x 55 mm, IP65 von RND Components entschieden. Das gibt es u.a. bei Reichelt in 3 verschiedenen Ausführungen. Ich habe mich für die günstigste Form entschieden, da ich weder in das Gehäuse reinschauen möchte, noch der Kasten ungeschützt aufgebaut wird.

💡 Übersicht

RND – Artikelnummer	455-00143	455-00132	455-00154
Werkstoff	ABS	PC	PC
Wesentliche Werkstoff-eigenschaften	In einem etwas kleineren Temperaturbereich einsetzbar als PC. -40 und +85 °C Geringfügig weniger UV-beständig im Vergleich zu PC. Für die Verwendung im Freien geeignet, wenn es vor Wettereinflüssen geschützt ist.	In einem großen Temperaturbereich einsetzbar. -40 und +130 °C Gute UV-Beständigkeit durch Beimengung von Stabilisatoren möglich.	
Einsatzempfehlung	z.B. im Schuppen	Im Freien sollte die direkte Sonneneinstrahlung vermieden werden.	
Deckelfarbe	Grau	Transparent	Grau
Preis (+Versandkost.)	~9 €	~14 €	~ 11 €

💡 Höhenprofile Raspberry Pi-HAT-LM2596 (XL4015)



⚠️ Das Gehäuse ist nicht für das Aufstecken des WittyPi geeignet.

Der XL4015 baut ca. 2 mm zu hoch, wenn dieser gesteckt werden soll.

Zum freien Positionieren, auch in anderen Gehäusen, gibt es z.B. Klebefüße mit M3 Innengewinde:

<https://www.horter-shop.de/de/111-selbstklebefuss-m3>

Alternativ können auch Stehbolzen, Muttern oder ein 3D gedruckter Fuß, z.B. mit 2K



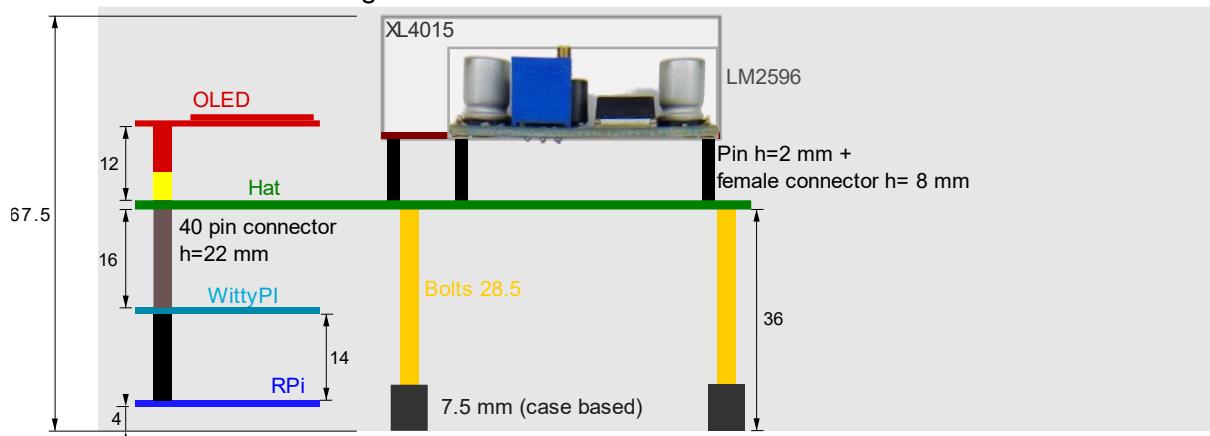
HoneyPi

Klebstoff eingeklebt werden. Hierbei sind die Kleberichtlinien (sauber, fettfrei, ggf. anrauen, usw.) zu beachten.

3D gedruckter Fuß und Selbstklebefuß:



Bei Einsatz eines WittyPi ist ein Gehäuse mit $>/ =75$ mm Höhe bzw. mit einem Innenmaß von mind. 69 mm notwendig.



💡 Kabel & Stecker

Ich selbst bevorzuge PG-Verschraubungen. Zwar ist im einen oder anderen Fall eine Steckverbindung praktischer, jedoch empfinde ich jede zusätzliche Steckverbindung als Störquelle. Gerade, wenn die Anlage ein paar Winter hinter sich hat. Wer Steckverbinder einsetzen möchte, sollte die Notwendigkeit einer UV-Beständigkeit prüfen bzw. in seiner Anlage bedenken und diese ggf. geschützt aufbauen.

→ Kunststoff-PG-Verschraubungen sind üblicherweise nicht UV beständig.

Von mir eingesetzte Baugrößen: PG11, PG9

Wer einen Satz Stecker kaufen möchte kann sich diesen Satz ansehen:

[Amazon.de/RUNCCI-YUN-8Pcs GX12-6 Pin, 12 mm](https://www.amazon.de/RUNCCI-YUN-8Pcs-GX12-6-Pin-12-mm/dp/B00VQKJLWY)

💡 Kabel für die Spannungsversorgung

Je nach eigener Auslegung ca. $0,5 \text{ mm}^2 - 2 \text{ mm}^2$

→ Für die Auslegung gibt es Rechner im Internet.

💡 Datenleitungen

HX711-Platine 4x $\sim 0,34 \text{ mm}^2$ (AWG22) bzw. mind. Cat 5

HX711-Wägezelle: 4x $\sim 0,34 \text{ mm}^2$ (AWG22) geschirmt, mind. Cat 5

Ich verwende Sensorleitungen, die aus einem anderen Projekt über geblieben sind. 8 adrige Netzwerkkabel CAT 5e, 6 oder 7 sind sehr günstig und gut verfügbar. Diese gibt es auch im Baumarkt für den Außenbereich(!).



HoneyPi

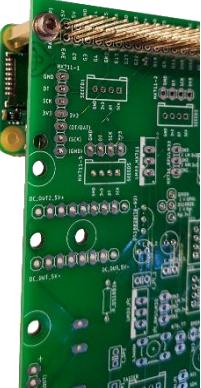
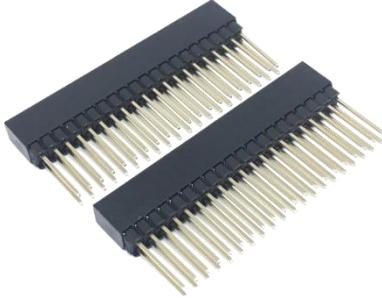
Die max. Länge, bzw. die Auswirkung, kann über verschiedenste im Internet befindliche Rechner ermittelt werden.



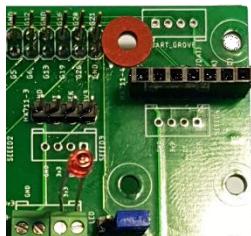
HoneyPi

8. Raspberry Pi Anschluss (40 pol.)

40 Pins für den Raspberry Pi

	Stacking Header (Bild zeigt 2 Stück)			
Gibt es mit h = ~19 mm oder h = ~22 mm				

 Im Bereich Raspberry bitte Isolierscheiben  unter den Schrauben verwenden.



Maße der Scheiben: (Innen Ø: 3,0 mm (2,5 mm), Außen Ø: 8,0 mm, Dicke: 0,5 mm
(Zum Beispiel INL 59915)

Stehbolzen verwende ich M2,5 x 14 mm.

→ Der Raspberry wird nur an der Platine verschraubt und „hängt“ in der Luft.

Ministückliste Stacking Header

Menge	Bezeichnung	LINK
1	Stacking Header	Aliexpress/ Stacking Header 20x2
1 Set	Stehbolzen und Schrauben	Aliexpress/ M2,5 8 mm + 6 mm =14 mm altern. 8 mm + 8 mm =16 mm
1 Set	INL 59915	Reichelt/ Isolierscheiben

Aus Kostengründen (~16 €) nutze ich den Raspberry Pi Zero. Entsprechend ist diese Platine auch dafür gemacht.

9. Gewichtsmessung

HX711 A/D-Wandler für die Wägezellen

Die Platine ist für den direkten Aufbau von zwei verschiedenen Typen von kl. Platinen mit dem A/D-Wandler vorbereitet (grün/violett), die sich in der Pin-Reihenfolge unterscheiden. Auf der Platine sind daher 2x 7 Pin-Holes vorhanden. Diese beiden Plätze können auch wie die weiteren drei HX711 Anschlüsse über Steckverbindungen angeschlossen werden. Somit



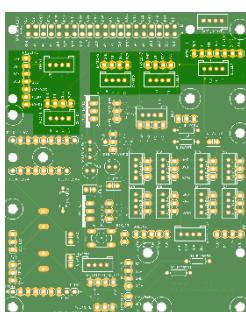
HoneyPi

sind 5 HX711 anschliessbar. Bei einem direkten Aufbau muss für eine Abstützung gesorgt werden.

Einen direkten Aufbau empfehle ich nicht! Das hat zwei Gründe: 1. Sollte der HX711 möglichst direkt an der Wägezelle sein und 2. habe ich beim Einsatz eines UMTS-Sticks vermehrt Störungen auf dem analogen HX711 Signal bekommen. Diese ließen sich mit einem Oszilloskop gut sichtbar machen. Sie waren auch auf den digitalen Signalen zu sehen, spielten da naturgemäß jedoch keine große Rolle.

Ich habe das nicht weiter untersuchen können. Mir reichte es, dass die Störungen ohne UMTS-Stick weg waren 😊

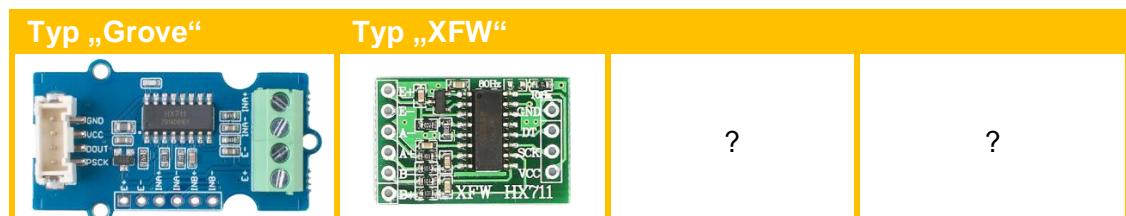
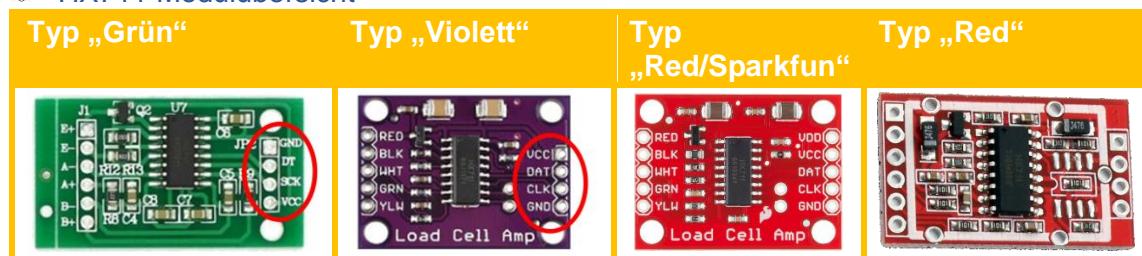
Daraus ist das „Sensor Board HX711 und DS18B20“ entstanden.



⚠ Anschluss, Kabel und Abschirmung

Die analoge Seite (HX711-Wägezelle) ist störungsanfälliger als die digitale Seite (HX711-Raspberry). Also sollte die analoge Seite kurz und gut abgeschirmt sein. Entsprechend habe ich eine Sensor Platine entwickelt. Diese Platine kann durch ein kleines Aluminiumgehäuse den HX711 gut abschirmen und direkt unter der Wage platziert werden. Diese Teilprojekt wird unter den Namen „HoneyPi Sensor Board“ geführt.

⚠ HX711 Modulübersicht



Die Bezeichnung über die Farbe bezieht sich auf die hier gezeigten Module. Die Hersteller können die Farbe frei bestimmen und so gibt es z.B. Modul Typ „Violett“ auch in der Farbe rot. Zu beachten sind der Pinreihenabstand, die Positionierung (vgl. rote Kreise) und die Pinbelegung.



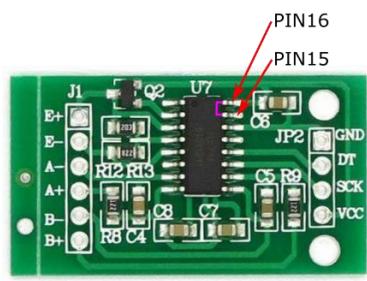
HoneyPi

ⓘ Typ „Grün“ analog GND(E-) und digital GND verbinden.

Im Netz gibt es verschiedene Meinungen dazu. Wenn der Widerstand zwischen analog GND(E-) und digital GND nicht gegen Null geht und es zu Kurz- oder Langzeitabweichungen oder auch zu starken temperaturabhängigen Abweichungen kommt, dann kann man analog GND(E-) und digital GND verbinden.

ⓘ Typ „Grün“ Sampling-Rate erhöhen

Manchmal klappt das Auslesen des HX711 per Honey Pi nicht oder nicht zuverlässig. Dann kann das Erhöhen der Frequenz von 10 auf 80 Hz eine Verbesserung bringen. Dazu muss man den Pin 15 des Chips von der Platine löten (Erwärmen und mit einer Nadel anheben) auf diesen Pin legt man dann VCC des Chips. → VCC sollte an Pin 16 anliegen und gut verwendbar sein.



ⓘ Typ „Violett“

Dieser Typ soll am besten funktionieren.

Bestellt, eingetroffen aber noch nicht probiert.

ⓘ Typ „Red Sparkfun“

Nicht getestet

ⓘ Typ „Red“

Die Belegung auf der Raspberry Pi Seite ist wie beim Typ „Violett“. Die andere Seite leider nicht. Die Möglichkeit der Abdeckung (Schirmung) finde ich interessant.

Ein kl. Test mit/ohne Abdeckung läuft gerade.

ⓘ Typ „Grove“

Vorteil ist hier, dass eine Grove-Buchse bereits aufgelötet ist.

ⓘ Typ „XFW“

Hat bei mir die größten Gewichtsschwankungen gezeigt. Ich habe nur ein Modul dieses Typs ausprobiert. Durch umlöten des kleinen Widerstandes (s. Bild, oben rechts) kann hier von 10 auf 80 Hz „umgeschaltet“ werden.

ⓘ Wägezelle

Ich benutze die Bosche H30A. Wer eine andere Wägezelle nutzen möchte, kann dies gerne tun. Ich empfehle dann, die Datenblätter zu vergleichen.

Für uns kommen nur Voll- und Halbbrücken-Wägezellen in Frage.

Die Vollbrücken-Wägezellen besitzen bessere Eigenschaften wie Linearität, höhere Empfindlichkeit sowie eine systematische Kompensation von Störeinflüssen wie Temperaturdrift und Kriechen. Wer vier Halbbrücken-Wägezellen einsetzen möchte sollte sich dessen bewusst sein.



HoneyPi

⚖️ Wägezelle und EMV-Schutz

Am besten wäre es, die Wägezelle samt Elektronik in ein verschlossenes und abschirmendes Metallgehäuse einzubauen. Das geht leider nicht. Daher ist die Abschirmung gut aufzulegen (Metallgehäuse für den HX711 + IRIS-Verschraubung). Das Kabel von der Wägezelle zum HX711 sollte abgeschirmt, kapazitätsarm und möglichst kurz sein.

Eine Wägezelle mit nicht abgeschirmtem Kabel würde ich nicht einsetzen.

🌡️ Temperaturdrift

Eine temperaturabhängige Gewichtsänderung von ca. 9 g/°C habe ich festgestellt. Das lässt sich für jedes System einfach ausmessen. Dabei ist der HX711 das temperaturempfindlichste Teil, vorausgesetzt man setzt eine Vollbrücken-Wägezelle ein.

10. Temperatursensor DS18B20 (1Wire)

Hierfür werden ein Widerstand mit 4,7 kΩ* und die entsprechenden Sensoren benötigt. Für die Mainboardtemperatur kann ein DS18B20 in der Bauform TO92 und für die Beute o.ä. ein Sensor im Edelstahlgehäuse verwendet werden.

DS18B20 - Edelstahlgehäuse:



DS18B20 - TO92:



Die Version 2.01 hat hier Anschlüsse für 1x Mainboard-Temperatur + 8x Anschlüsse für den DS18B20 im Edelstahlgehäuse (Grove, Klemmstein oder PIN/Dupont).

*Bei einer grösseren Menge und auch in Abhängigkeit der Topologie kann ein kleinere Widerstand (2,2 kΩ) notwendig sein.

Die Stern Topologie bzw. die Baum Topologie ist am häufigsten im Einsatz und bei vielen Bus-Teilnehmern schwieriger in den Griff zu bekommen, da hier Reflexionen entstehen können. Bei Busabstürzen kann ein Keramik-Kondensator (100 nF / 3v3 - GND, nicht an DATA!) vor/an jedem Sensor Abhilfe schaffen.

Da zum Zeitpunkt des Platinen Layouts die Ursache für die Busabstürze nicht geklärt war, haben wir uns für verschiedene Methoden der Spannungsversorgung für die DS18B20 Sensoren entschieden.

3,3 V vom
Raspberry Pi

3,3 V vom GPIO09

3,3 V vom LF33CV (HX711)



HoneyPi

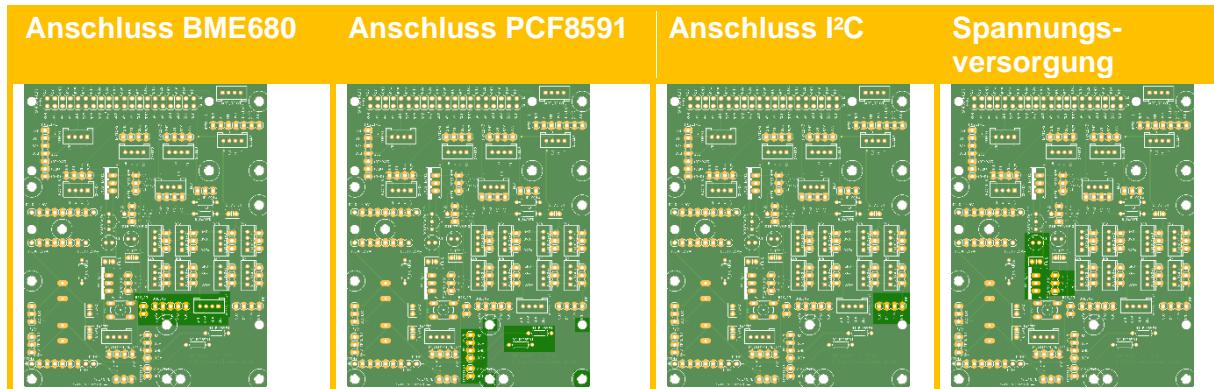
Raspberry Pi	Per Software schaltbar	Raspberry Pi unabhängig 3,3 V / vom LF33CV

Ministückliste DS18B20

Menge	Bezeichnung	LINK
1-?	DS18B20 - Edelstahlgehäuse	Aliexpress/ DS18B20 - Edelstahlgehäuse
1	DS18B20 - TO92	Aliexpress/ DS18B20 - TO92
1	Widerstand 4,7 kΩ	

11. I²C Schnittstellen / Sensoren

Übersicht

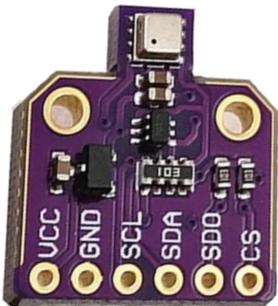


BME680

Die Version 2.01 bietet die Möglichkeit zum direkten Einsatz von einem Sensor. Da es sich hier um einen I²C Bus handelt, gibt es hier Überlegungen diese Schnittstelle auch anderweitig zu nutzen. Der BME680 verwendet die Adressen 0x76 oder 0x77. Beim Anschluss über die RM 2,54 mm Pin-Holes kann per Jumper / Lötbrücke SD0 auf GND gelegt werden. Somit verändert sich die Adresse und der Einsatz eines 2. BME680 ist über den Grove Stecker oder den I²C Pin-Anschluss möglich. Das setzt voraus, dass bei Deinem Modul die Adressänderung so erfolgen kann. Es gib auch Module, da muss 3v3 auf den SD0 gelegt werden und/oder auf dem Modul eine Leitung getrennt werden.



HoneyPi



Hinweis: CS ist nicht zu belegen!

⚠ BME280

Die Version 2.01 bietet die Möglichkeit zum direkten Einsatz von einem Sensor anstelle des BME680. Der BME280 verwendet die Adressen 0x76 und 0x77(Lötbrücke / SD0*).

Typ „Lötbrücke“	Typ „SD0“
	

Zur Adressänderung ist hier die Brücke (rot) zu trennen und eine Brücke (blau) zu erzeugen.

Zur Adressänderung ist hier SD0 mit GND zu verbinden.

* Ich hatte ein Modul (Bild oben, rechts), wo die Adressänderung, anders als beim BM680, nicht funktionierte.

⚠ 4 Pin Module / Lötbrücke BME280

Das Sensormodul mit 4 Pins besitzt eine Lötbrücke von SDO zu GND. Dies führt zur I²C-Adresse 0x76.

Unterbricht man diese Lötbrücke durch Aufkratzen mit einem scharfen Messer und verbindet das mittlere Lötpad mit dem dritten Lötpad, führt dies zu einer Verbindung von SDO mit VIN. Dies führt zur I²C-Adresse 0x77.

Quelle: <https://forum-raspberrypi.de....>

Siehe auch: [bme280](#)

⚠ Ob die Adressänderung mit 3,3 V oder GND erreicht werden kann muss ausgemessen werden, bevor man den SD0 anschließt.

⚠ Ministückliste BME280/BME680

Menge	Bezeichnung	LINK
1	BME680	Aliexpress/ BME680
1	BME280	Aliexpress/ BME280 3,3v



HoneyPi

1

Ggf. Grovebuchse oder
JST PH2.0

⚠ Auf die Bus-Adressen achten.

⌚ Alternative I²C Sensoren

ATH10

HDC1008 ($\pm 4\%$ & $\pm 0.2^\circ\text{C}$)

SHT31-D ($\pm 2\%$ & $\pm 0.3^\circ\text{C}$)

Einen schönen Vergleich einiger Sensoren findest Du hier:

http://www.kandrsmit.org/RJS/Misc/Hygrometers/calib_many.html

12. SHT20 (nicht kompatibel mit SHT3x / noch nicht vollständig umgesetzt)

Den SHT gibt es auf einer kleinen Platine um das man ein Gehäuse aus einigen Kaufteilen bauen kann. Der max. Durchmesser beträgt dann 18 mm. Die Platine selbst ist nur 7x20mm groß.



Die Kombination aus Sensor, PG7 Verschraubung, 6-Kantmuffe und Schalldämpfer* gibt es bei Aliexpress komplett zu kaufen. Es gibt auch andere, ähnliche Ausführungen.



*Der Schalldämpfer stammt aus der Pneumatik und dient da zur Dämpfung der Abluftgeräusche. Hier wird das poröse, also luft- und feuchtigkeitsdurchlässige, Material als „Schutzgitter“ genutzt.

Möge die fleißige Biene dieses Teil nicht zukleistern 😊

→ Bei Bedarf können einfach kl. Löcher gebohrt werden.

Der SHT20 verwendet die I²C Bus-Adresse 0x40

⚠ Alternativ gibt es auch diese Gehäuse mit SHT20 nur leider war bei meiner Bestellung ein HTU20 enthalten und der funktioniert nicht.



Bei längeren Strecken kann ein Bus-Extender, wie der P82b715, verwendet werden.

→ Siehe „I²C Bus: Verbindungsaufläufe, o.ä. Probleme“



HoneyPi

13. I²C Adressenübersicht

Sensor	I ² C- Addr.	Sens- type	Beute No.	Temp	Humidity	Pressure	Air Quality	CO2	UV (A/B)	Light	Voltage	Special
GPS (Pimoroni PA1010D)	0x10										x	
CJMCU-2317 (MCP23017)	0x20	-									x	
BH1750	0x23 (0x5c)	11								x		
OLED Display	0x3C	-									x	
AHT10												
AHT15												
AHT20	0x38 oder 0x39	10	1, 2	x	x							
AHT21												
HDC1080												
HDC2080												
SI7021	0x40 oder 0x41	8										
not yet			3, 4	x	x							
HTU21D												
SHT25	0x40	12		x	x							
SHT30												
SHT31												
SHT35	0x44	9	5	x	x							
SHT85												
PCF8591	0x48	6								x		
ee895	0x5C	7						x				
Si1145 (Gy1145)	0x60	not yet							x	x		
WittyPi	0x68+0x69										x	
SHTC1 (CJMCU-189)	0x70	not yet	not yet	x	x							
SHTC3												
BME280	0x76 oder 0x77	5	6, 7	x	x	x						
BME680		1					x					

⚠️ I²C Adresserweiterung

Wer weitere I²C Sensoren, wie z.B. BME680 (mehr als 2) verwenden möchte, kann das Modul TCA9548A dazwischenschalten und die Adressen anpassen. Dann muss auch in der HoneyPi Software eine Anpassung erfolgen. **Dieser Ansatz wurde erstmal verworfen.**





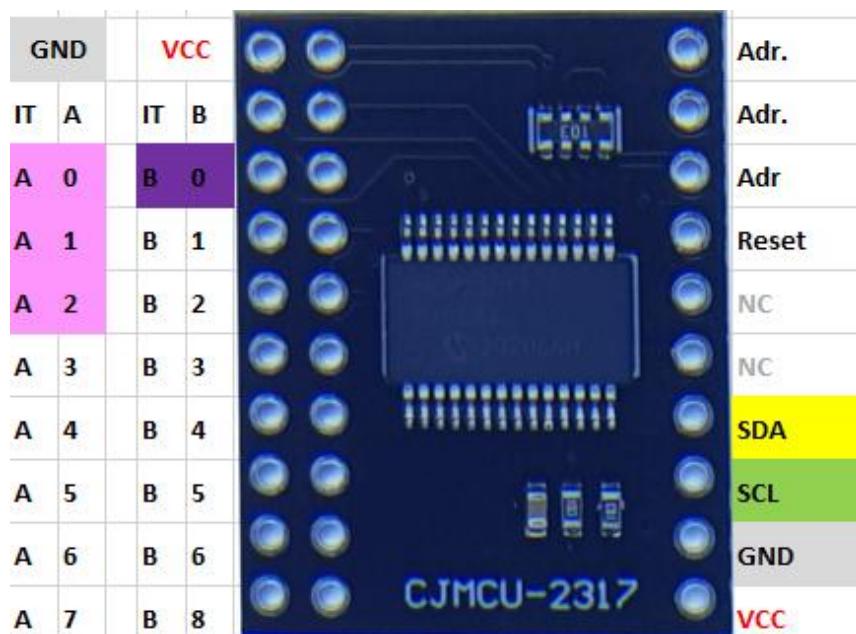
HoneyPi

14. Mehr GPIO Ports (CJMCU-2317) für die Menüsteuerung des OLED

Mehr GPIO-Ports kann man über den Chip MCP23017 erhalten, den es als Modul CJMCU-2317 gibt. Das Modul wird über den I²C-Bus angesteuert. Hier können einfache Bauteile wie Taster und Schlüsselschalter angeschlossen werden.

Für die geplante Menüsteuerung (in Kombination mit dem Display) werden die Ausgänge A0, A1 und A2 verwendet. Der Schlüsselschalter ist an B0 vorgesehen.

Indem alle drei Adr. Leitungen mit GND verbunden werden, wird die Bus-Adresse 0x20 eingestellt.



15. Kleinteile

Auflistung ist noch offen. Bitte erstmal in den Schaltplan gucken.

16. Sensor Board HX711 und DS18B20

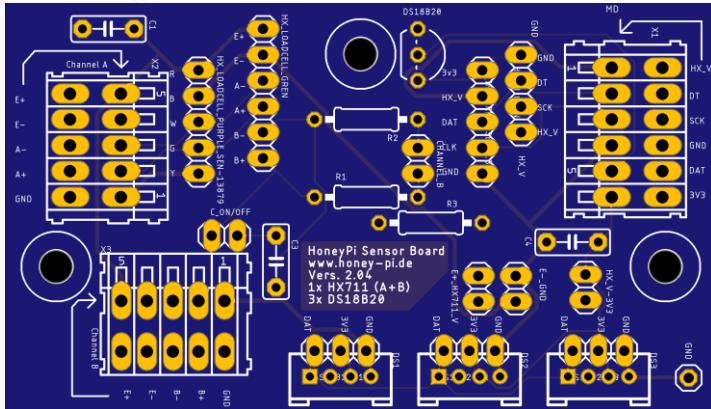
Um die analoge Leitungslänge möglichst kurz und mit dem HX711 weit weg von UMTS-Sticks o.ä. zu sein, ist dieses Board entstanden. Es wird je ein Board pro Beute gebraucht, wenn nur der A-Kanal verwendet werden soll.

Dieses Board kann an die Platine 2.x, 3.x und auch ohne eine Platine direkt an den Raspberry Pi angeschlossen werden.

Eine Darstellung des Sensor Board 2.04:



HoneyPi



Ein Foto von der Vorgängerversion im Gehäuse* Rose 01.06 10 03:



Das abgebildete Gehäuse war bereits mit PG9 Gewindebohrungen ausgestattet und ich habe es daher so eingesetzt.

PG-Verschraubungen sind abhangig vom Kabelauendurchmesser zu whlen.

Hinweis: 3x PG9 passt nur mit geschnittenem Gewinde an der langen Gehäuseseite. Die PG-Muttern bauen grösser!

→ Alternativ PG7 oder M12 prüfen und nutzen.

*Beim Einsatz der Federkraftklemmen von Wago 233-506, die später hinzugekommen sind, kann das Gehäuse etwas knapp werden. Diese Klemmen wurden auf Wunsch hinzugefügt. Ich persönlich setze diese nicht ein. Bitte beachte, dass nur die mit Pfeil gekennzeichneten Pin-Reihe belegt ist.

Wago 233-506:

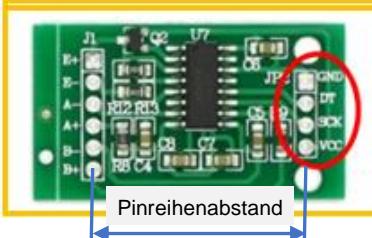


Einsetzbare Module

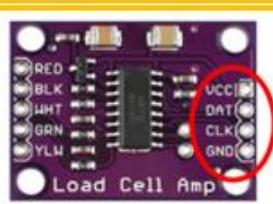


HoneyPi

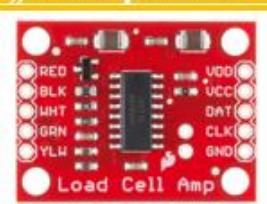
Typ „Grün“



Typ „Violett“



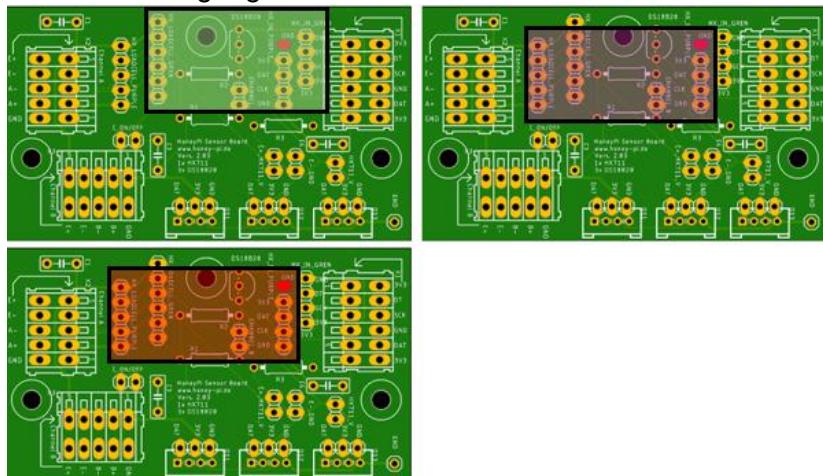
**Typ
„Red/Sparkfun“**



Typ „Grün“ hat einen Pinreihenabstand von 10 Pins ($10 \times 2,54 \text{ mm} = 25,4 \text{ mm}$).

Typ „Violett“ und „Rot/Sparkfun“ (ab vers. 2.04) haben einen Pinreihenabstand von 11 Pins ($11 \times 2,54 \text{ mm} = 27,94 \text{ mm}$).

Auf die Pinbelegung ist zu achten!



Auf dem Board sind Jumper und Kondensatoren setz- bzw. auflötbar.

Wir bitten diese nur in Abstimmung, z.B. über Facebook, zu nutzen.

Der auflötbare DS18B20 dient zur Temperaturmessung und Korrektur des Gewichtes.

Weiter Infos sind dem Schaltplan auf GitHub zu entnehmen.



HoneyPi

17. Kabelverschraubung Tabelle

PG Kabelverschraubung Tabelle (PG-Durchmesser Tabelle)

Nenngröße	Gewinde Außendurchmesser	Gewinde Steigung	Durchmesser Bohrung	Kabeldurchmesser
PG 7	12,5 mm	1,27 mm	12,8 mm	3,0 - 6,5 mm
PG 9	15,2 mm	1,41 mm	15,5 mm	4,0 - 8,0 mm

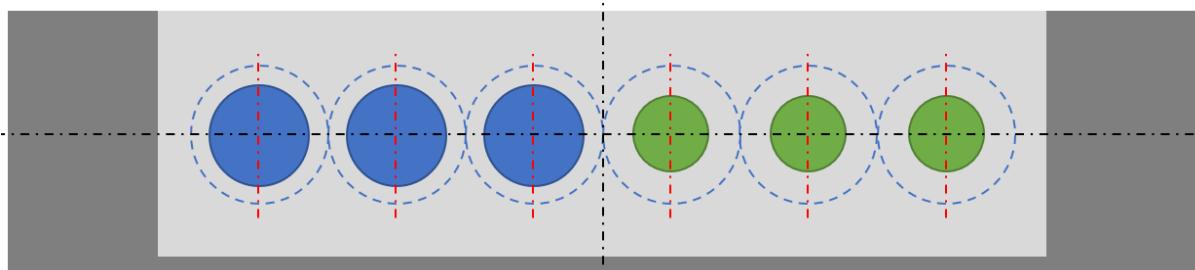
Metrische Kabelverschraubung Tabelle

Nenngröße	Gewinde Außendurchmesser	Gewinde Steigung	Durchmesser Bohrung	Kabeldurchmesser
M8 x 1,0	8 mm	1,0 mm	8,5 mm	3,0 - 5,0 mm
M10 x 1,0	10 mm	1,0 mm	10,5 mm	3,0 - 6,0 mm
M12 x 1,5	12 mm	1,5 mm	12,5 mm	3,5 - 7,0 mm
M16 x 1,5	16 mm	1,5 mm	16,5 mm	4,5 - 10,0 mm

Es gibt herstellerabhängige Abweichungen!

Muster Bohrbild

Bei dem verwendeten Gehäuse können seitlich 6x M16 (auch 6x PG9) Kabelverschraubung gesetzt werden. Die Größe der Kabelverschraubung ist vom Kabelaußendurchmesser und vom Hersteller abhängig. Alle gesetzten Gehäuseöffnungen sollten nach unten zeigen.



Darstellung zeigt 3x M16 + 3x M12 als Beispiel. Alle Bohrungen sind für einen Abstand für M16 (Durchmesser des Kreises mit blauer Strichpunktlinie = 22 mm).

→ Es lassen sich so also 6x M16/PG9 Verschraubungen gut setzen.

18. Software Konfiguration

Um beim Raspberry auch ohne angeschlossenen Monitor etwas prüfen oder ändern zu können, bietet sich der SSH Zugang an. Ich verwende dazu unter Windows PUTTY bzw. unter Android JuiceSSH.

⚠ Bei der Erstinbetriebnahme sollte immer ein Monitor angeschlossen werden!

19. Tabelle Verbraucher

Pos.	Menge	Bezeichnung	Spannung [V]	Strom[mA] (Menge =1)	Strom[mA] (Menge =n)	Am RPi (3,3 V)
1	1	Raspberry Pi Zero	5	170		
2	1-5	HX711	3,3	1,5		
3	1	LED Typ: Rot, 2V, 10mA, Vorwiderstand: Rv =130 Ohm	2 (an 3,3)	10		
4	1-8	DS18B20	3,3	1,5		
5	1-2	BME280	3,3	0,05		
6	1-2	BME680	3,3	12		



HoneyPi

7	1	ATH10	3,3				
8	1	SHT31	3,3				
9	1	SI7021	3,3				
10	1	HTU21D	3,3				
11	1	ee895	3,3				
12	1	HDC1080	3,3				
13	1	OLED	5				
14	1	WittyPi	5	7			
15	1	PCF8591	3,3	2,81			
16	1	USB / UMTS Modem Huawei E303	5	750 (Peak ~2000)			
17		USB / UMTS Modem Huawei E3531i-2	5	700 (Peak ~2000)			
18	1	LM2596	12	5-10			
19		XM4015	12	max. 30			
20		LF33CV	5				

Um den Gesamtstrombedarf zu erfassen, muss die Summe aller Ströme 5 V und 3,3 V gebildet werden. Nun wird entweder auf 12 V (beim Einsatz des LM.../ XM...) oder 5 V (bei direkter 5 V Versorgung) umgerechnet.

20. Beispiel für eine Stromüberschlagsrechnung

Summe des 1. Stromkreises (3,3 V):

150mA (Zusammengezählt gem. der „Tabelle Verbraucher“ / 3,3 V)

$$P_1 = U \times I_{3,3} V$$

$$P_1 = 3,3 V \times 0,15 A$$

$$P_1 = 0,495 W$$

Die Leistung = 0,495 W wird nun auf 12 V umgerechnet:

$$I_{12V} = P_1 / U$$

$$I_{12V} = 0,495 W / 12 V$$

$$I_{12V} = 0,04125 A$$

Summe des 2. Stromkreises (5 V):

920mA (Zusammengezählt gem. der „Tabelle Verbraucher“ / 5 V)

$$P_2 = U \times I_5 V$$

$$P_2 = 5 V \times 0,92 A$$

$$P_2 = 4,6 W$$

Die Leistung 4,6 W wird nun auf 12 V umgerechnet:

$$I_{12V} = P_2 / U$$

$$I_{12V} = 4,6 W / 12 V$$

$$I_{12V} = 0,3833 A$$

Nun können wir die Ströme aus den beiden Stromkreisen addieren:



HoneyPi

$$\begin{aligned}I_{\text{Ges.}} &= I_{12V(P1)} + I_{12V(P2)} \\I_{\text{Ges.}} &= 0,04125 \text{ A} + 0,3833 \text{ A} \\I_{\text{Ges.}} &= \underline{0,4125 \text{ A (@12 V)}}\end{aligned}$$

Wenn Du die Möglichkeiten hast, kannst Du diesen Wert durch eine Messung des Stromverbrauchs prüfen. Dabei wirst Du feststellen, dass selten dieser Strom fließen wird. Das liegt zum Beispiel daran, dass nicht die ganze Zeit gemessen wird und erst nach der Messung übertragen wird.



HoneyPi

21. Belegungstabelle 40pol. / GPIO / PIN / Device

Belegung der GPIOs über den 40 Pol. Steckverbinder zur Platine

Platine	Raspberry	Platine
Unbelegt	3,3 V Power	5 V Power
I2C-BUS	GPIO2 SDA1 I2C	5 V Power
I2C-BUS	GPIO3 SCL1 I2C	Ground
WittyPi	GPIO4	GPIO14 UART0_TXD
GND	Ground	GPIO15 UART0_RXD
WittyPi	GPIO17	GPIO18 PCM_CLK
HX711_DT_(3)	GPIO27	Ground
HX711_SCK_(3)	GPIO22	GPIO23
3,3V (Taster, diverse)	3,3 V Power	GPIO24
Unbelegt	GPIO10	Ground
1-Wire DS18b20 3,3V	GPIO9	GPIO25
1-Wire, DS18b20 Data	GPIO11	GPIO8 SPI0_CE0_N
Unbelegt	Ground	GPIO7 SPI0_CE1_N
RESERVED	I2C ID EEPROM	I2C ID EEPROM
HX711_DT_(1)	GPIO5	Ground
HX711_SCK_(1)	GPIO6	GPIO12
HX711_SCK_(2)	GPIO13	Ground
Unbelegt	GPIO19	GPIO16
HX711_SCK_(5)	GPIO26	GPIO20
Unbelegt	Ground	GPIO21



HoneyPi

22. Belegungstabelle GPIO

GPIO	Belegung	Nr.	Bemerkung
2	I ² C SDI/SDA		reserviert
3	I ² C SCL/SCK		reserviert
4	WittyPi		reserviert
5	HX711 DT	1	Standardbelegung der HoneyPi Software
6	HX711 SCK	1	Standardbelegung der HoneyPi Software
7	nicht belegt		keine Standardzuordnung in der Honey Pi Software
8	nicht belegt		keine Standardzuordnung in der Honey Pi Software
9	DS18B20 3.3V		Standardbelegung der HoneyPi Software
10	nicht belegt		keine Zuordnung in der Honey Pi Software
11	DS18B20		Standardbelegung der HoneyPi Software
12	HX711 DT	2	Standardbelegung der HoneyPi Software
13	HX711 SCK	2	Standardbelegung der HoneyPi Software
14	UART TX		Standardbelegung der HoneyPi Software
15	UART RX		Standardbelegung der HoneyPi Software
16	Button		Standardbelegung der HoneyPi Software
17	WittyPi		reserviert
18	nicht belegt		Standardbelegung der HoneyPi Software: MAX CLK
19	nicht belegt		Standardbelegung der HoneyPi Software: MAX MISO
20	nicht belegt		keine Standardzuordnung in der Honey Pi Software
21	LED		Standardbelegung der HoneyPi Software
22	HX711 SCK	3	Standardbelegung der HoneyPi Software
23	HX711 DT	4	Standardbelegung der HoneyPi Software
24	HX711 SCK	4	Standardbelegung der HoneyPi Software
25	HX711 DT	5	Standardbelegung der HoneyPi Software
26	HX711 SCK	5	Standardbelegung der HoneyPi Software
27	HX711 DT	3	Standardbelegung der HoneyPi Software

23. Teilebeschaffung und Einkaufsliste

Einen Tipp: Erst gut planen, dann bei wenigen Händlern bestellen und somit Versandkosten sparen. Mit Aliexpress.com habe ich gute Erfahrungen gemacht, wenn die Lieferung etwas dauern darf. Die Teile sind hier so günstig, dass auch die Bestellung bei verschiedenen Händlern sinnvoll sein kann. Man benötigt dafür allerdings eine Kreditkarte und sollte unter ~25 EUR pro Lieferung (gilt für Deutschland) liegen.

- Die geforderten Steuerabgaben wären für mich IO gewesen. Leider musste ich bei einer Bestellung zum Posthauptzollamt und dort fast 4 Stunden auf eine Bearbeitung warten.

Christian W. war so nett und hat eine Einkaufsliste bei Reichelt erstellt:

- HoneyPi Platinen Bestückung: <https://tidd.ly/3iBj30E>

Javan hat für die Platine v2 diese [strukturierte Projektstückliste](#) erstellt: [Einkaufsliste](#)
Zudem gibt es noch die „alte“ Einkaufsliste auf Honeypi.de selbst: <https://www.honey-pi.de/Einkaufsliste/>

Hier sind „Affiliate-Links“ enthalten, die das Projekt Honeypi.de unterstützen.



HoneyPi

24. Ausblick

Neben einer Version für die großen Raspberry Pis, wie 3 und 4, denke ich gerade über die Erweiterbarkeit nach. Daraus ist auch das [kl. Sensorboard](#) für 1x Hx711 und 3x DS18B20 entstanden. Auch soll diese Anleitung weiterwachsen.



HoneyPi

25. Bauanleitung(en)

💡 Vorbereitung oder „Wie fange ich an“?

Also, wie würde ich anfangen 😊? Nun, der erste Schritt wäre, dass ich mir Gedanken machen würde, was ich mit HoneyPi messen möchte. Daraus ergibt sich dann eine Liste der notwendigen und zu bestellenden Teile.

- ➔ Ich gehe hier davon aus, dass Du löten kannst und das notwendige Equipment hast. Wenn nicht, findest Du Anleitungen im Internet oder auf www.youtube.com. Vielleicht gibt es ja in Deinem Bekanntenkreis jemand, der Dir hilft?
- ➔ Vielleicht lötest Du auch alle Teile auf die Platine, um nur „1x löten“ zu müssen und bei Veränderungen draußen am Stock, dann nur noch stecken oder klemmen zu müssen.

Bevor Du bestellst, solltest Du Dir Gedanken zum zeitlichen Ablauf machen. Die Bestellung der Platine bei <https://ilcpcb.com/> dauert ca. 10-15 Tage. Hast Du Zeit und möchtest Geld sparen, dann bietet sich eine Bestellung der Komponenten bei Aliexpress.com, Banggood.com o.ä. an. Auch über Ebay kann man Produkte direkt aus Asien bestellen. Du wirst leider nicht alles bei einem Händler vorfinden. Trotzdem solltest Du versuchen zusammenzufassen und so Versandkosten zu sparen. Vielleicht bestellst Du auch mit jemanden, z.B. aus dem Forum, zusammen.

Ich würde grob so vorgehen:

- Bedarfsanalyse
- Funktionsliste
- Materialliste
- Beschaffung
- Aufbau

💡 Bauvorschlag I

Bedarfsanalyse und Funktionsliste:

Sensoren: Gewichtsmessung mit 1x HX711, Temperaturmessung mit 2x DS20b18

Stromversorgung: 12 V, mit Spannungsregler LM2596 (steckbar)

💡 Materialiste und Beschaffung:

<https://docs.google.com/spreadsheets/d/1J12ud60raVa6e5kRixf1TCkPQCki4IKJRttvNYuWrT8/edit>

💡 Aufbau:

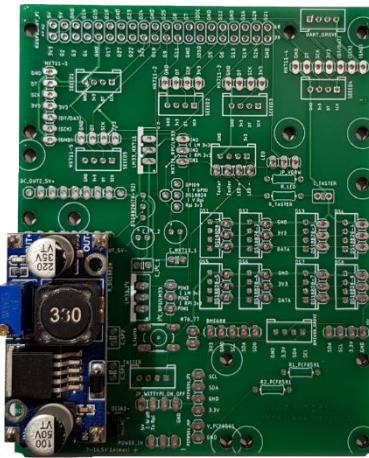
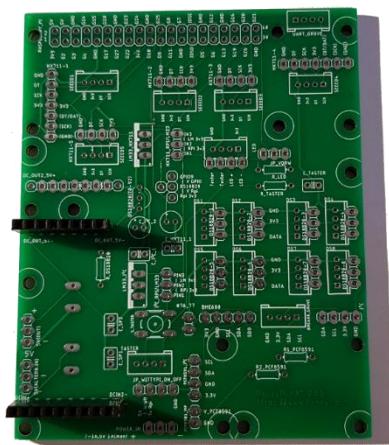
Zum Aufbau benötigst Du eine Buchsenleiste, die Du zurechtsägen oder einfach mit dem Seitenschneider zurechtschneiden kannst. Da ich optional das größere Modul XL4015, anstatt des LM2596 aufbauen möchte, kommt unten die 9 pol. Buchsenleiste und oben eine 8 und 9 pol. Buchsenleiste hin.

- ➔ Beim Zerschneiden opfere ich einen Pin der Buchsenleiste, da ich in der Mitte z.B. des 10. Buchsenkontakte schneide. Wer möchte feilt dann das Ende plan.
- ➔ Beim Auflöten helfen in die Buchsenleiste gekreuzt eingesteckte Pinleisten.



HoneyPi

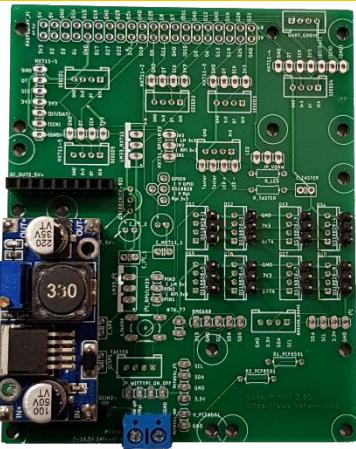
LM2596



Als Spannungsanschluss habe ich einen Klemmblock mit 5.08 mm Pin-Abstand gewählt, da der zu meinem Kabelquerschnitt passt. Hier kann aber auch ein kleiner 2.54 mm Klemmbock gesetzt werden.

- Von den 3 Pins sind 2 mit GND verbunden.

12 V in



Jumper: Raspberry Pi / WittyPi:

Ohne WittyPi muss hier gebrückt werden. Mit Pins und Jumper oder Draht und Lötzinn. Alternativ kann man hier auch den WittyPi „dazwischen klemmen“.

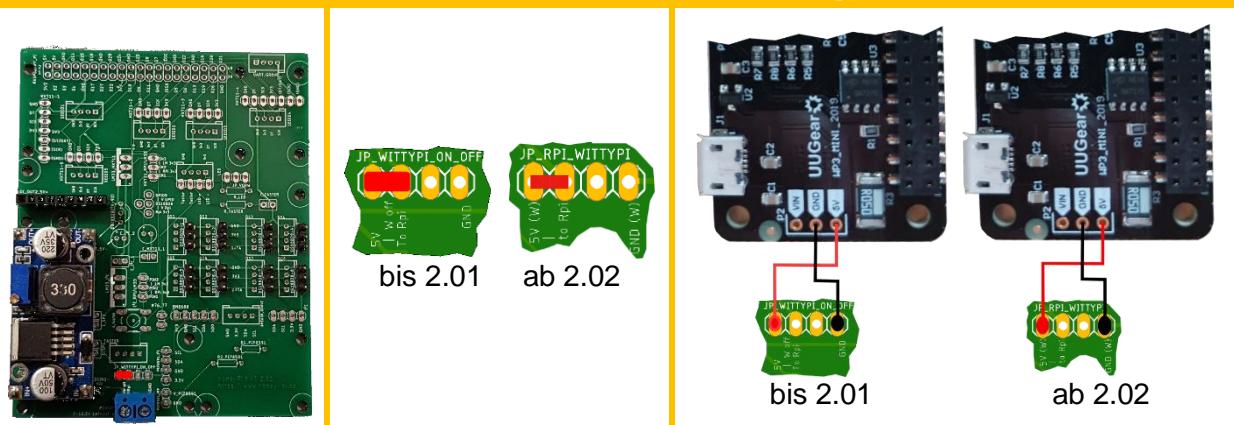


HoneyPi

Jumper Raspberry Pi / WittyPi

Jumper / Brücke Stromversorgung ohne WittyPi

Anschluss WittyPi (der Jumper auf der Platine entfällt, s. vereinfachte Darstellung)



Beim Einsatz des WittyPi wird die Brücke nicht gesetzt. Hier werden nur die 5 V, die z.B. von LM2596 / XM4015 o.ä. kommen, von der Platine abgegriffen.

Mit Version 2.02 wurde die Bezeichnung verändert.

Bis einschl. Version 2.01:

5 V Platine → WittyPi 5 V
GND Platine → WittyPi GND

Ab Version 2.02:

5 V(W) Platine an WittyPi 5 V
GND(W) Platine an WittyPi GND

- ⚠ Der WittyPi muss natürlich auch auf den 40pol. Steckverbinder, zwischen Raspberry und Platine, gesteckt werden.
- ⚠ Dieser Teil der Anleitung bezieht sich auf den WittyPi 3 mini.
- ⚠ Beim Einsatz eines WittyPi ist es sinnvoll einen weiteren Taster zu installieren um den „Schlafmodus“ ausschalten und den Raspberry Pi starten zu können. Dazu muss am Pin „Switch“ und Pin „GND“ der Taster angeschlossen werden. Beide Pins sind am 'Unpopulated 7-Pin Header (P3)' des WittyPi zu finden. Weiteres bitte der Anleitung zum WittyPi entnehmen.

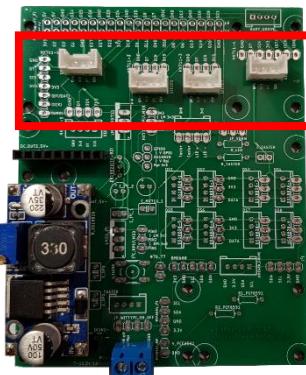
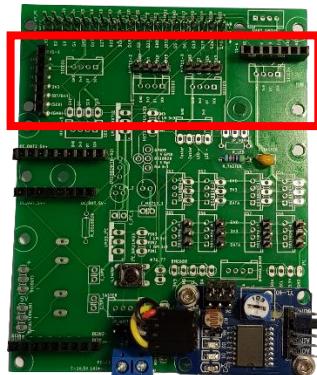
HX711:

Auf Wunsch gibt es neben dem Rastermaß 2,54mm auch das passende Rastermaß für die Grove-Buchsen.



HoneyPi

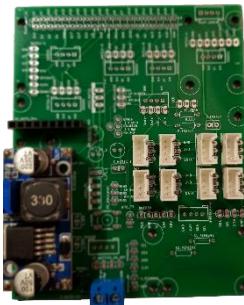
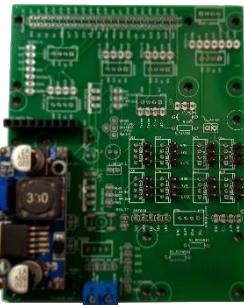
HX711



DS18B20

2,54 mm Raster für 8x Pins oder 6x 3er Klemmblöcke (nur 6 Stück, da der 7 und 8 Anschluss schlecht zugänglich ist). Für Grove-Liebhaber 8 Anschlüsse.

DS18B20



Raspberry Pi Anschluss

Wenn das mit dem Löten gerade so gut klappt 😊, könnte man gleich die 40 Pins für den Raspberry anlöten. Ich verwende "Board-zu-Board Steckverbinder" / Stacking Header 2x20.

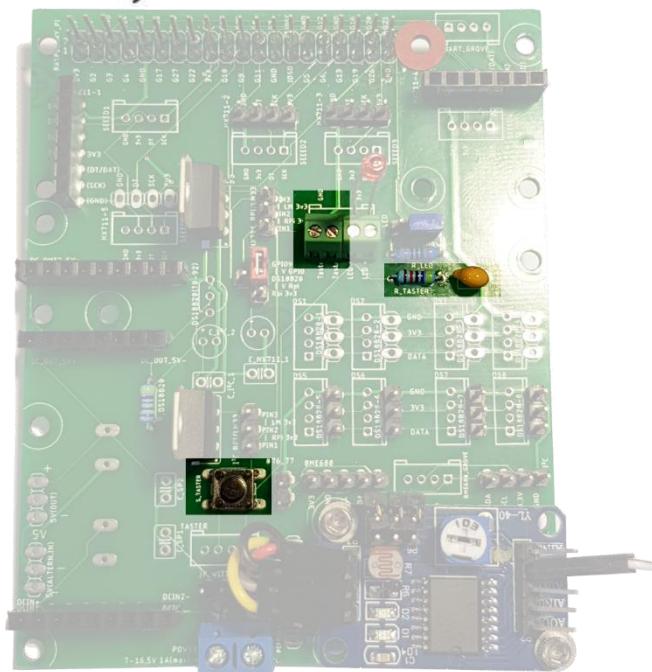
- ⚠ Wenn man diese Platine auf einen Raspberry Pi 3 oder 4 aufsetzen möchte und die Bauhöhe und Orientierung keine Rolle spielen, dann steckt man einen 2. Stacking Header als Höhenadapter auf und kommt so über die USB & LAN Anschlüsse des Raspberry Pi.

Taster

Es gibt hier 3 Anschlussmöglichkeiten. Für alle die keinen Gehäusetaster wünschen, kann hier ein kleiner Taster aufgelötet werden. Hierfür ist auch die Entprellung ausgelegt. Bei einem anderen Taster muss diese ggf. angepasst werden. Grove und Klemmstein sind die Alternative. Die Taster können auch parallel angeschlossen werden. Ich habe einen Widerstand von 50 kΩ und einen Keramikkondensator mit 22 nF genommen.



HoneyPi



LED

Zum Testen und Zeigen habe ich hier an einen Klemmblock eine LED (10 mA) angeschlossen.

→ Das lange Beinchen an der LED ist an Plus anzuschließen 😊.

Vorwiderstand: $150\ \Omega$ (rechnerisch 130 Ohm).

Der Jumper muss rechtspositioniert* werden, damit der Strom durch den Vorwiderstand fließt.

*Ab der Version 2.01 ist der Jumper nur notwendig, wenn der Widerstand gebrückt werden soll. Gebrückt werden kann der Widerstand für LEDs, die mit einem Vorwiderstand ausgerüstet sind.

40 Pins für den RASPBERRY PI

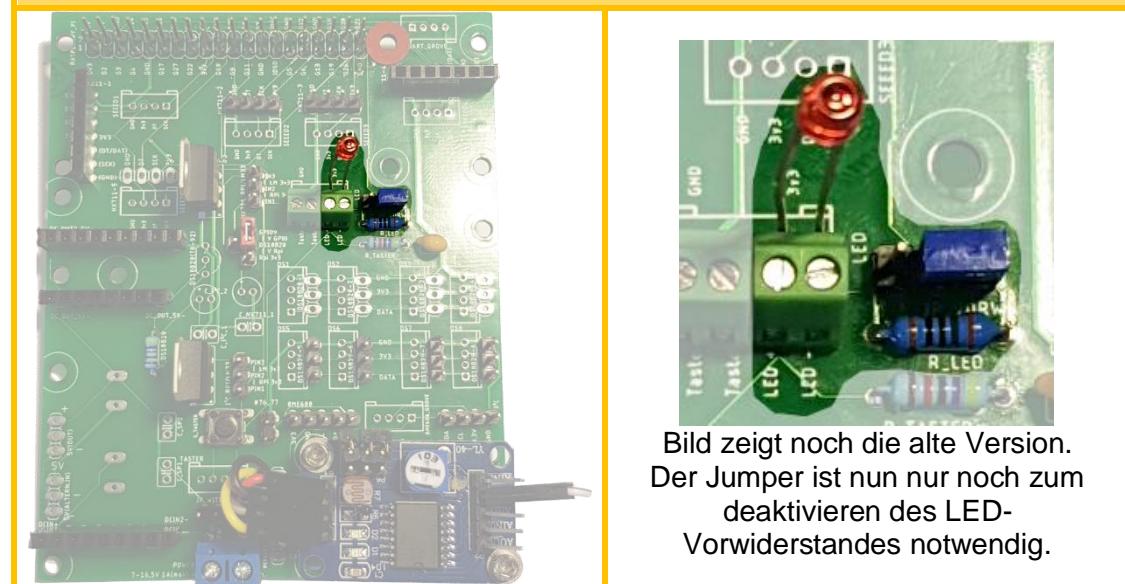


Bild zeigt noch die alte Version.
Der Jumper ist nun nur noch zum
deaktivieren des LED-
Vorwiderstandes notwendig.



HoneyPi

26. Trouble-shooting guide / FAQ / Tipps

🛡 Keine Spannung?

Gehe bei der Fehlersuche systematisch vor:

- ▶ Messe mit dem Multimeter an verschiedenen Punkten.
- ▶ Ausgehend von der Spannungsquelle, dann nach den Spannungswandlern.
- ▶ Nutze den Schaltplan. Dieser ist stark vereinfacht und auch für Anfänger geeignet.

Mögliche Fehler:

- ▶ Jumper JP_WITTYPI_ON_OFF muss ohne WittyPi gesetzt sein.
Siehe: „Jumper Raspberry Pi / WittyPi“
- ▶ Sicherung: Ist eine Sicherung gesetzt?
F1 (12 V) oder F2 (5 V)
Siehe: Absicherung
- ▶ GND ist nicht gleich GND. Ein gut zugänglicher GND ist bei den Pins JP_WITTYPI_ON_OFF.

🛡 Raspberry bootet nicht von derr SD-Karte

SD-Karte beschrieben?

BelenaEtcher unter Windows mit Adminrechten probiert?

LEDs? Wie blinken die LEDs des Raspberry Pis

Monitor angeschlossen? Ausgabe?

🛡 Taster geht nicht

🛡 Sensoren gehen nicht / bei Steckverbindern.

🛡 HX711

Im Log erscheint eine Fehlermeldung analog zu dieser hier:

WARNING – HoneyPi.read_hx711 | HX711 DT: 12 SCK: 13 Channel: A **46.43%, in total 19 of 41 elements removed by filter within hx711**. You might need to check your power supply or cabling setup.

Ist das ein Problem?

Der HX711 ist sehr empfindlich und digitalisiert das Signal, dass der analoge Teil bis zur Wägezelle zur Verfügung stellt. Das Signal ist bei den „preiswerten half-bridge Wägezellen etwas schlechter als bei den full-bridge Wägezellen wie z.B. der Bosche H40A. Dazu kommen der Aufbau und die Kabellänge des analogen Teils (s.o.). HoneyPi Verwendet daher eine statistische Methode um die „besten“ Messwerte zu Mitteln und Ausreißer zu verwerfen. Auch unter den besten Bedingungen werden noch Messwerte verworfen werde. Ein guter erreichbarer Wert ist um 7/41. Das bedeutet das 7 Messwerte „ab vom Schuss“ sind und nicht berücksichtigt werden.

🛡 Uhrzeit

Im Logfile sind Einträge die mit „**Timestamp -...**“ beginnen zu finden.

Timestamp bedeutet Zeitstempel und das die Uhrzeit hier fehlt.



HoneyPi

Der Raspberry Pi hat keine Hardwareuhr (RTC = Real Time Clock) und verliert die Uhrzeit durch das Trennen der Spannungsversorgung. Daher bezieht die HoneyPi Software die Uhrzeit über einen Netzwerkverbindung von einem Zeitserver.

ⓘ Generelles zum I²C Bus (Links):

http://www.netzmafia.de/skripten/hardware/RasPi/RasPi_I2C.html

<https://rn-wissen.de/wiki/index.php/I2C>

<http://www.elektronik-magazin.de/page/der-i2c-bus-was-ist-das-21>

<https://www.i2c-bus.org/i2c-primer/typical-i2c-bus-setup/>

ⓘ I²C Bus Teilnehmer prüfen:

Dazu muss man an der Kommandozeile des Raspberry diese Zeile eingeben:

i2cdetect -y 1

OK, SD0 auf GND:

```
pi@HoneyPi:~ $ i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- 76 --
pi@HoneyPi:~ $
```

OK, SD0 ist nicht auf

GND

```
pi@HoneyPi:~ $ i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- 77 --
pi@HoneyPi:~ $
```

Nicht (richtig) angeschlossen:

```
pi@HoneyPi:~ $ i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- --
```



HoneyPi

⚠ I²C Bus Verbindungsausfälle, o.ä. Probleme:

Der I²C Bus ist so spezifiziert, dass die Kapazität der Busleitung nicht über 400 pF (besser nur 200 pF) sein darf.

Ein Cat.5 Kabel hat eine Kapazität von 17 pF pro Fuß (@100MHz, 1 Fuß = 30,48 cm) umgerechnet 0,56 pF/cm also 56 pF/m. Es wären also 7,14 m theoretisch (eher 5 m) möglich. Theoretisch? Ja, denn leider gibt es noch andere Einflussgrößen. Auch nehmen es nicht alle so genau mit der Spezifikation 😊 .

Spätesten dann sollte man einen Bustreiber wie den P82B715 einsetzen oder ggf. ein besseres Kabel probieren. Alternativ könnte man die Übertragungsfrequenz (100 Mhz) heruntersetzen.

Stattdessen werde ich mich an einer kleinen Platine mit dem P82B715 Bustreiber probieren. Damit sollen ca. 50 m möglich sein, wenn das Problem konkret wird.

Einfacher ist es, die Übertragungsfrequenz zu reduzieren, denn das soll ca. 30 m ermöglichen.

Das sind zwei Lösungsansätze die von mir noch nicht ausprobiert wurden.

⚠ I²C Bus: Pullups reduzieren

CMOS-Eingänge wie die GPIO-Eingänge des Raspberry Pi neigen dazu, zufällig in die eine oder andere Richtung zu schalten. Also high = 3,3 V oder low = 0 V zu werden und das würde zu unerwünschten Effekten führen. Daher werden Pullup-Widerstände eingesetzt. Die programmierbaren Pullup-Widerstände des Raspberry Pi's liegen bei ca. 50 kΩ (GPIO 2 und GPIO 3: 1,8 kΩ, fest aktiviert).

Beim I²C-Bus können alle Geräte den Bus auf GND ziehen. Keines der Gerät am I²C Bus kann den High-Pegel (auf SDA oder SCL) wieder herstellen. Um einen High-Pegel auf dem Bus zu erzeugen, ist ein Pullup-Widerstand notwendig. Fehlt der Pullup-Widerstand ist keine Datenübertragung möglich.

Einige I²C Module haben einen solchen Widerstand auf dem Modul verbaut. Schaltet man jetzt mehrere dieser Module zusammen, so sind alle diese Pullups parallel geschaltet. Hat man mehrere solcher Module, dann wird der resultierende Widerstand niedriger.

Die Lösung wäre dann denkbar einfach: Die Pullups müssen auf einigen Modulen deaktivieren werden.

Die GPIO 2 und 3 (für den I²C) haben 1,8 kΩ Pullup-Widerstände (fest aktiviert).

⚠ Auf dem Raspberry Pi sind die Pullups aktiv.

⚠ I²C Bus Stromverbrauch und Pullups:

Im Internet gibt es Bedenken aufgrund des Stromverbrauches beim Einsatz mehrerer Pullup-Widerstände am Bus. Rechnerisch ist sind diese als Widerstandsparallelschaltung zu betrachten.

Viele Module haben 10 kΩ, einige 4,7 kΩ Pullup-Widerstände.

Der Widerstand R1 ist im folgenden Beispiel der Pullup des Raspberry Pis und R2 der des angeschlossenen Moduls.

Beispiel Rechnung 1:

(R1 = 1,8 kΩ; R2 = 10 kΩ)

Rges. = R1 x R2 / (R1 + R2)

Rges. = 1,8 kΩ x 10 kΩ / (1,8 kΩ + 10 kΩ)

Rges. = 18 kΩ / (11,8 kΩ)



HoneyPi

Rges. = 1,52 kΩ

$$I = U/R_{\text{ges.}}$$

$$I = 3,3 \text{ V} / 1525,4 \text{ Ohm}$$

$$I = 0,00226 \text{ A}$$

I = 2,26 mA → das ist nicht viel, fließt aber „immer“.

Beispiel Rechnung 2:

$$(R_1 = 1,8 \text{ k}\Omega; R_2 = 4,7 \text{ k}\Omega)$$

$$R_{\text{ges.}} = R_1 \times R_2 / (R_1 + R_2)$$

$$R_{\text{ges.}} = 1,8 \text{ k}\Omega \times 4,7 \text{ k}\Omega / (1,8 \text{ k}\Omega + 4,7 \text{ k}\Omega)$$

$$R_{\text{ges.}} = 8,46 \text{ k}\Omega / (6,5 \text{ k}\Omega)$$

$$R_{\text{ges.}} = 1,301 \text{ k}\Omega$$

$$I = U/R_{\text{ges.}}$$

$$I = 3,3 \text{ V} / 1301,5 \text{ Ohm}$$

$$I = 0,00253 \text{ A}$$

I = 2,53 mA → das ist nicht viel, fließt aber „immer“.

Bei 5 Teilnehmer ($R_1 = 1,8 \text{ k}\Omega; R_2-5 = 4,7 \text{ k}\Omega$) komme ich auf $R_{\text{ges.}} = 711 \text{ }\Omega$ und einen Strom von 4,641 mA und bei 10 Teilnehmer ($R_1 = 1,8 \text{ k}\Omega; R_2-10 = 4,7 \text{ k}\Omega$) $R_{\text{ges.}} = 405 \text{ }\Omega$ und einen Strom von 8,148 mA.

Selbst bei 20 Teilnehmern ($R_1 = 1,8 \text{ k}\Omega; R_2-20 = 4,7 \text{ k}\Omega$) 373 Ω ist der Strom nur bei 8.847 mA. Das ist nicht viel.

⚠️ I²C Bus: Übertragungsfrequenz reduzieren & Reichweite erhöhen

In die config.txt einfügen:

dtparam=i2c_arm=on,i2c_arm_baudrate=32000

-> Neustart

Zum Anzeigen der Geschwindigkeit an der Konsole sind diese drei Zeilen einzugeben:

```
var ="$(xxd /sys/class/i2c-adapter/i2c-1/of_node/clock-frequency | awk -F' ' '{print $2}')"
```

```
var =${var//[:blank:]\}/}
```

```
printf "I2C Clock Rate: %d Hz\n" 0x$var
```

⚠️ Geht nicht ;-(

Ich helfe gerne aber mit „geht nicht“ kommen wir nicht weiter.

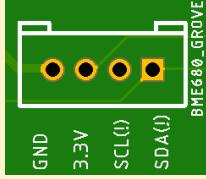
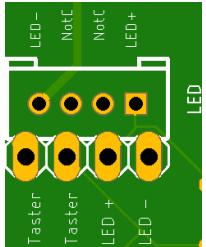
Wenn ich helfen soll, dann musst DU meine Augen sein. Ein paar scharfe(!) Fotos und eine Fehlerbeschreibung sind ein guter Anfang.

→ In der [Facebook Gruppe \(https://www.facebook.com/TheHoneyPi/\)](https://www.facebook.com/TheHoneyPi/) gibt es viele User, die auch helfen können.



HoneyPi

27. Änderungshistorie Platine

Version	Datum	Änderung
P2.00	01.11.2020	Erster Entwurf, nicht öffentlich.
P2.01	27.09.2020	<ul style="list-style-type: none">- Diverse kleine Fehlerkorrekturen.- LED Jumper vereinfacht.- Grove-Anschluss UART 90° gedreht.
P2.01.1	04.01.2021	<p>BME680_GROVE Pinbelegung fehlerhaft. Auf Pin 1 soll SCL und Pin 2 SDA liegen. Die Belegung ist verdreht.</p> <p>Workaround: Für Platinen bis einschl. 2.01 ist die Beschriftung anzupassen und Schaltplan 2.01.x zu verwenden. Die Pinbelegung ist im Stecker anzupassen.</p> 
		<p>Platine 2.02 ff wurde angepasst. Hier gilt der Schaltplan 2.02</p>
P2.02	04.01.2021	<ul style="list-style-type: none">- BME680_GROVE Pinbelegung korrigiert.- Bezeichnung NotC (not Connected) bei der LED-Grovebuchse hinzugefügt. 



HoneyPi

28. Änderungshistorie des Dokuments

Version	Datum	Änderung
D2.01	27.09.2020	Erster Entwurf
D2.02	20.10.2020	HX711 „red“ ergänzt. HX711 Sensor Board HX711 + DS18B20
D2.02	06.11.2020	Änderungen im Bereich BME680/280 zu SD0 Sensor Board HX711 und DS18B20 hinzugefügt. Änderungshistorie: Versionsnummern um D (für Dokument) und P (für Platine) erweitern.
D2.03		Einige Teilelisten mit Links eingefügt. Belegungsliste 40pol. hinzu. GPIO Belegungsliste hinzu Taster Entprellen korrigiert. Hinweise zur Kabellänge, EMV und zur Wägezelle hinzugefügt.
D2.03		Hinweis hinzu: Zusätzlichen Taster für den WittyPi einbauen.
		Tabelle Kabelverschraubungen hinzu
D2.04	18.12.2021	I ² C Adressen
		Sensor SHT20
		Kleine Korrekturen, Aufräumarbeiten.
D2.05	29.12.2021	GPIO Erweiterung CJMCU-2317
D2.06		Musterbohrbild eingefügt. Div, kleine Korrekturen, Aufräumarbeiten.
D2.07	29.04.2022	Sensorboard HX711 2.04 eingefügt Div, kleine Korrekturen, Aufräumarbeiten.