



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 Honey-Pi.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.Honey-Pi.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 3/4. Wenn ein WittyPi mini eingesetzt wird, kann auch ein Raspberry Pi ¾ eingesetzt werden.

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



Inhalt

1.	PLATINE V2.01	4
2.	FUNKTIONSLISTE DES BOARDS:	4
	• Spannungsversorgung und Messung	4
	• Taster (Wartungsmodus=an/aus)	4
	• LED	4
	• Gewichtsmessung	5
	• Temperaturmessung 1-Wire.....	5
	• Diverse I ² C Bus-Sensoren.....	5
	• WittPi.....	5
	• Timer Board	5
3.	SPANNUNGSVERSORGUNG.....	6
	• Übersicht	6
	• LM2596 Modul (Einspeisung 12 V)	6
	• XL4015 Modul (Einspeisung 12 V).....	7
	• Einspeisung 5 V	7
	• Versorgung der Sensormodule mit Festspannungsregler LF33CV für 3,3 V	7
	• Muss ich einen Spannungsregler verwenden?	8
	• Kann ich auch einen anderen Spannungsregler als den LF33CV verwenden?.....	8
	• Ministückliste LF33cv	9
	• Brücken (Jumper) HX711	9
	• Brücken (Jumper) I ² C.....	9
	• Brücken (Jumper) DS18B20 (Spannungsversorgung).....	9
4.	ABSICHERUNG	11
5.	SPANNUNGSMESSUNG.....	12
	• PCF8591 (I ² C)	12
	• Spannungsteiler	13
	• Musterberechnungen: Spannungsteiler.....	13
	• Ministückliste PCF8591	14
6.	TASTER	14
	• Taster entprellen	14
	• Musterberechnungen: Grenzfrequenz f ₀	14
7.	LED.....	15
	• Vorwiderstand für die LED.....	15
8.	GEHÄUSE	16
	• Übersicht	16
	• Höhenprofile RPi-HAT-LM2596.....	16
	• Kabel & Stecker	17
	• Kabel für die Spannungsversorgung.....	17
	• Datenleitungen	17
9.	RASPBERRY PI ANSCHLUSS (40 POL.).....	18
10.	GEWICHTSMESSUNG	18
	• HX711 A/D-Wandler für die Wägezellen	18
	• Anschluss, Kabel und Abschirmung	19
	• HX711 Modulübersicht.....	19
	• Typ „Grün“ analog GND(E-) und digital GND verbinden.	19
	• Typ „Grün“ Sampling-Rate erhöhen.....	20



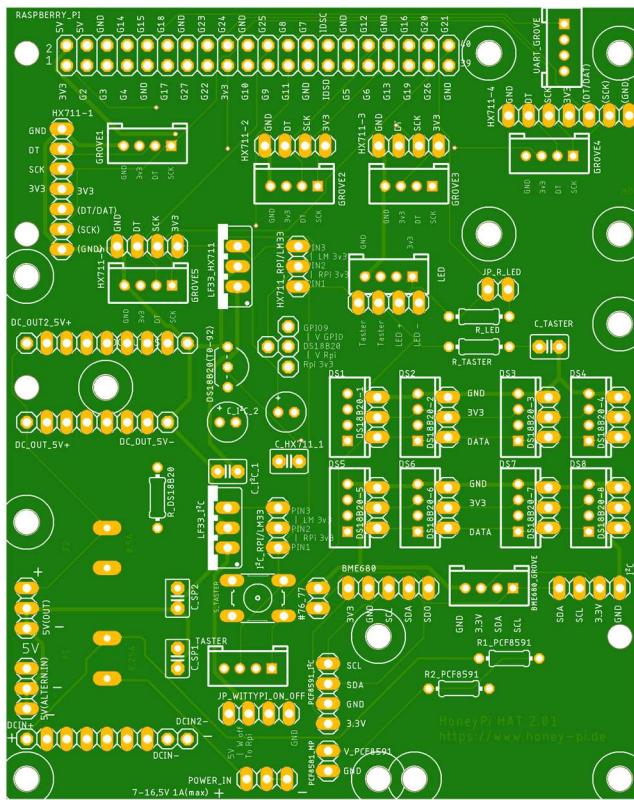
HoneyPi

• Typ „Violett“.....	20
• Typ „Red Sparkfun“.....	20
• Typ „Red“.....	20
• Typ „Grove“.....	20
• Typ „XFW“.....	20
• Wägezelle	20
• Wägezelle und EMV-Schutz	21
• Temperaturdrift.....	21
11. TEMPERATURSENSOR DS18B20 (1WIRE).....	22
12. I ² C SCHNITTSTELLEN / SENSOREN	23
• Übersicht	23
• BME680.....	23
• BME280.....	23
• 4 Pin Module / Lötbrücke BME280.....	24
• Alternative I ² C Sensoren (noch nicht in HoneyPi integriert).....	24
• I ² C Adressen.....	24
13. KLEINTEILE	25
14. SENSOR BOARD HX711 UND DS18B20	25
• Einsetzbare Module	25
15. TROUBLE SHOOTING.....	26
• HX711.....	26
• Generelles zum I ² C Bus (Links):.....	26
• I ² C Bus Teilnehmer prüfen:	26
• I ² C Bus: Verbindungsausfälle, o.ä. Probleme:	27
• I ² C Bus:Pullups reduzieren:.....	27
• I ² C Bus:Übertragungsfrequenz reduzieren:	28
16. SOFTWARE KONFIGURATION.....	28
17. TABELLE VERBRAUCHER.....	29
18. BEISPIEL FÜR EINE STROMÜBERSCHLAGSRECHNUNG	29
19. BELEGUNGSTABELLE 40POL. / GPIO / PIN / DEVICE	31
20. BELEGUNGSTABELLE GPIO	32
21. TEILEBESCHAFFUNG UND EINKAUFSLISTE.....	32
22. AUSBLICK.....	33
23. BAUANLEITUNG(EN).....	34
• Vorbereitung oder „Wie fange ich an“?.....	34
• Bauvorschlag I.....	34
• Aufbau:	34
24. ÄNDERUNGSHISTORIE PLATINE	39
25. ÄNDERUNGSHISTORIE DOKUMENT	39



HoneyPi

1. Platine v2.01



Die Platine 2.01 (Maße 100x80mm, 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](#) bereit.

Meine Platinen bestelle ich bei <https://jlpcpcb.com/>

2. 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

WittiPi Schnittstelle

5 V Schnittstelle für

2x I E33CV zur Erzeugung einer 3:

EX ER 300V zur Erzeugung einer 3,0 V Spannung aus der 9 V Spannung



Taster (Wartungsmodus=an/aus)
Board Taster (Printmontage) und S

Onboard Taster (Printmontage) und Steckverbinder (Tx RMZ.0 mm für Grove, Tx RMZ.54 mm für Pins, Klemmsteine o.ä.) für einen Gehäusetaster.



(blink)

RM2.0 mm für Grove

Dienstag 15. September 2020



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

Besser ist es, die fünf Schnittstelle zum Sensor Board 2.x 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 DS10b20 (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)

WittPi

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

Timer Board

Gedacht als einfache Lösung anstelle des WittPi.

Die Hardware funktioniert. Die softwareseitige Unterstützung durch Honey-Pi wurde noch nicht angegangen.



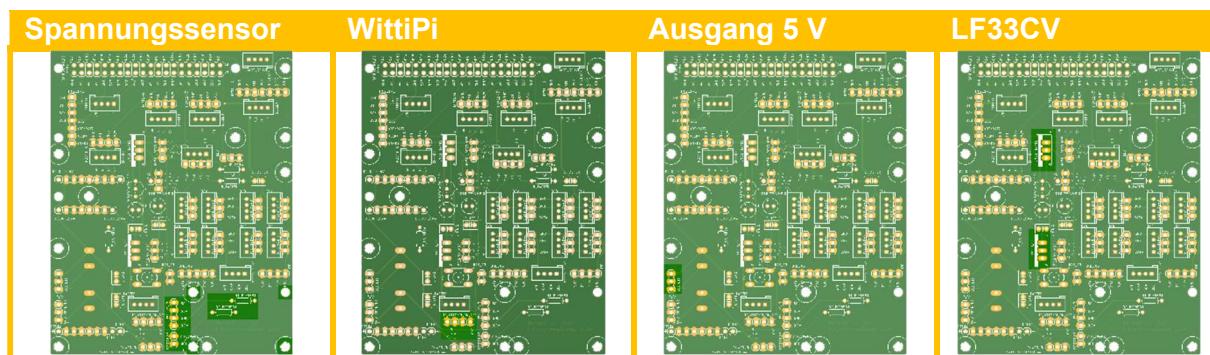
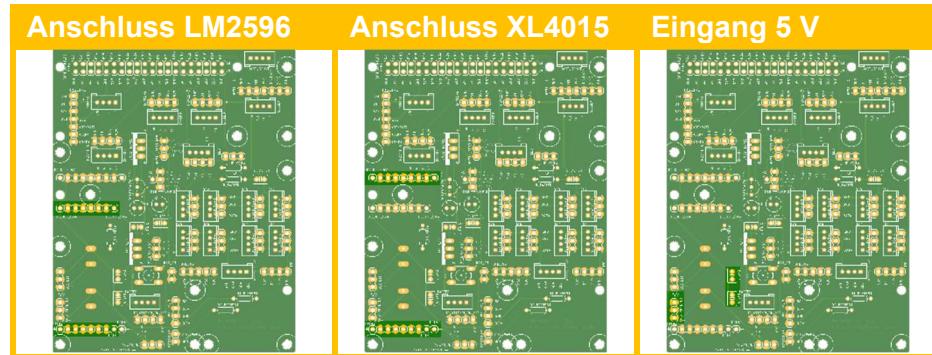
HoneyPi

3. 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.

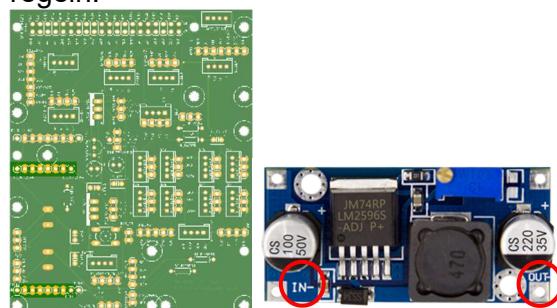
Es können hier die größeren RM 5,08 mm Klemmsteine und alternativ auch die kleineren 2,54 mm Klemmsteine eingesetzt werden. Dazu sind von den 3 Pin-Holes ein Pin-Hole für + und 2 für GND vorgesehen. Der kleine Klemmsteine kommt in die Pin-Holes 1 und 2 und der große Klemmstein in die Pin-Holes 1 und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

💡 Übersicht



💡 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!



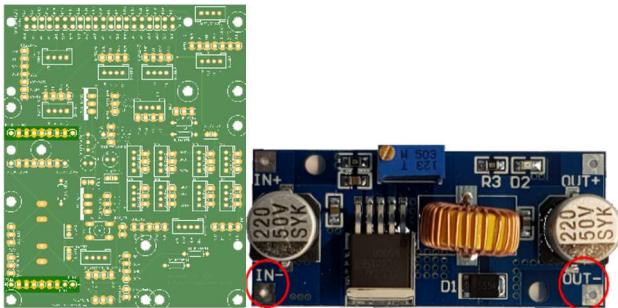
HoneyPi

⚠️ 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.

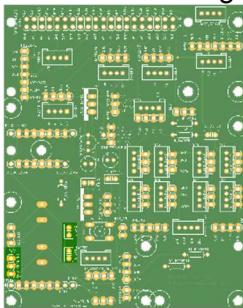
Im Heise.de Forum habe ich von „spontanen“ Ausfällen des LM2596 gelesen. Daher habe ich dieses alternative Modul vorgesehen. Auch wenn weitere Verbraucher geplant werden sollten, ist dieses Modul eine Alternative, da ein wesentlich höherer Strom bereitgestellt werden kann.



- ⚠️ 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. $\sim 1 \mu\text{F}$ und $100 \mu\text{F}$) 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 $22 \mu\text{F}$ wird das Schwingen auf ca. +/- 0,01 V reduziert. Das Datenblatt gibt eine mind. Kapazität von $2,2 \mu\text{F}$ vor und in Musterapplikationen wird ein $10 \mu\text{F}$ Elektrolytkondensator eingesetzt. Das Verhalten des $10 \mu\text{F}$ Elkos konnte ich nicht mehr visualisieren, da ich keinen



HoneyPi

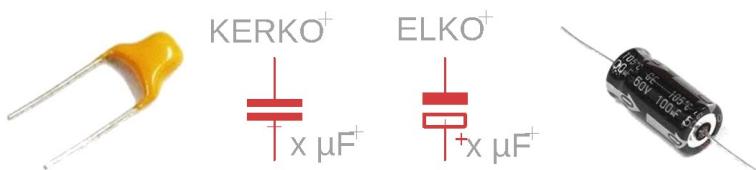
Zugriff auf ein Oszilloskop mehr habe. 10 µF und 22 µF sind in meinen beiden Aufbauten verbaut und laufen unauffällig/gut.

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-(li) oder Elektrolytkondensator(re):



Eingangsseitig ist ein 0,1 µF 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 mir 10mA alle zusammen mit max. 30 mA.

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

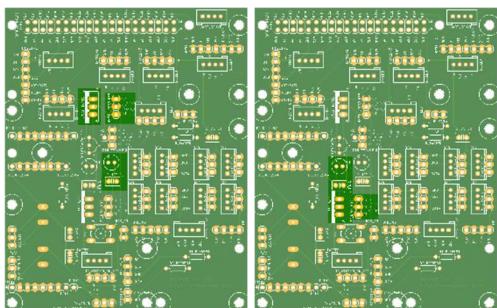
Klar, den musst Du dann allerdings selber auslegen. Neben der Pinbelegung können auch andere Eigenschaften anders sein.

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

Der Spannungsregler LF33 fällt nicht, wie ggf. andere Spannungsregler gleich aus, wenn die Eingangsspannung zu gering ist, sondern er liefert dann eine entsprechend geringere Ausgangsspannung. Beispielsweise sind bei 3,1 V Batteriespannung, ungefähr 2,8 V Ausgangsspannung zu erwarten.



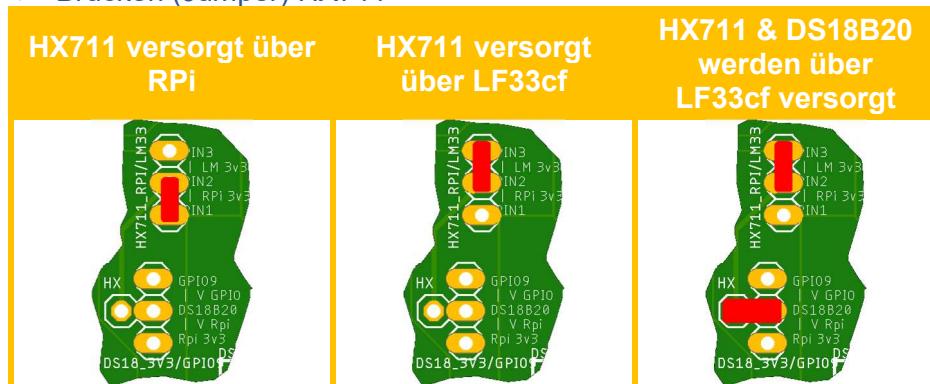
HoneyPi



❖ Ministückliste LF33cv

Menge	Bezeichnung	LINK
1-2	LF33cv	Aliexpress
1-2	Keramikkondensator 0,1 µF, >/= 5V	
1-2	Elektrolytkondensator 10 µF, >/= 5V	

❖ Brücken (Jumper) HX711



❖ Brücken (Jumper) I²C

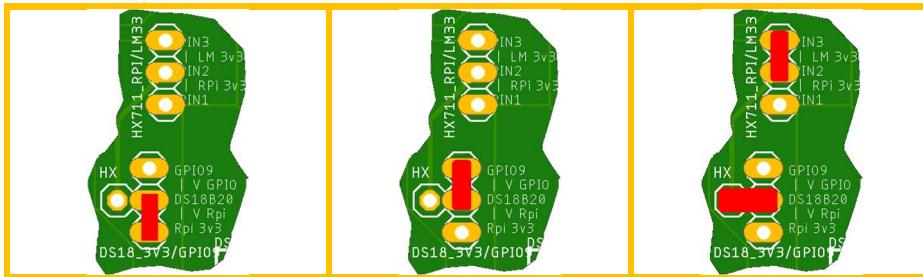


❖ Brücken (Jumper) DS18B20 (Spannungsversorgung)





HoneyPi

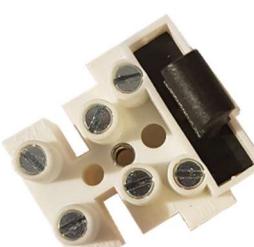




HoneyPi

4. 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 18V und die Batterie nur 12V“, 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)
	

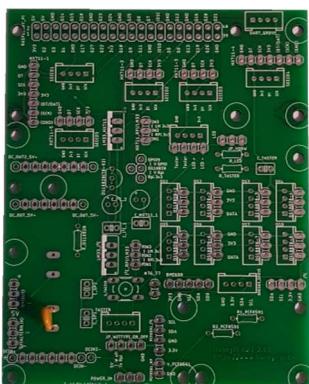
- ⚠ Sicherungen Lösen beim ~1,5-1,6-fachen des Nennstromes aus. Genaue Daten findest Du in den Datenblättern.

Zur Absicherung der Platine sind zwei Möglichkeiten zum Anschluss einer Sicherung vorhanden.
1x für die Eingangsspannung vor den Spannungsreglern (LM2596 o. XM4015) und 1x bei 5v Spannungsversorgung (nach der Spannungsquelle).



HoneyPi

Vor dem Spannungswandler



5V Anschluss (ohne Spannungswandler)



Ich nutze, mit dem Wissen das 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 0,5A Polifuse ein. Beim Einsatz eines UMTS Modems wird diese nicht reichen. Hier würde ich max. 1A ansetzen.

Bei 5V Anschluss / ohne Spannungswandler kann die 2. Position, mit einer Sicherung für eine zu bestimmende Stromstärke, genutzt werden.

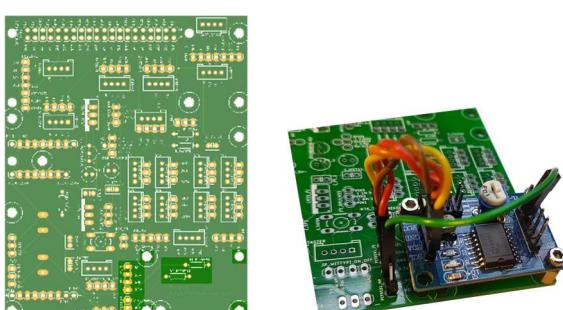
5. Spannungsmessung

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 Honey-Pi 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 $I = U/R$
$P = U^2/R$	$P = U^2/R$
$P = 16,5 \text{ V} * 16,5 \text{ V} / 37500 \Omega$	$P = 16,5 \text{ V} * 16,5 \text{ V} / 50.000 \Omega$
$P = 0,00726 \text{ W}$	$P = 0,005445 \text{ W}$
$I = P/U$	$I = P/U$
$I = 0,00726 \text{ W} / 16,5 \text{ V}$	$I = 0,005445 \text{ W} / 16,5 \text{ V}$
$I = 0,00044 \text{ A}$	$I = 0,00033 \text{ A}$
$I = 0,44 \text{ mA}$	$I = 0,33 \text{ mA}$



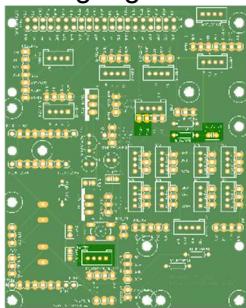
HoneyPi

Ministückliste PCF8591

Menge	Bezeichnung	LINK
1	PCF8591	Aliexpress
1	Widerstand 30 KOhm ($\leq 1\%$)	
1	Widerstand 7,5 KOhm ($\leq 1\%$)	
1 Set	Stehbolzen und Schrauben	Aliexpress

6. 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

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

$$f_0 = 1 / (2\pi \cdot 50.000 \text{ Ohm} \cdot 0.000.000.1 \text{ F})$$

$$R = 47 \text{ kOhm}$$

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

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

Default (OnBoard Taster)

$$R = 50 \text{ kOhm}$$

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

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

Menge	Bezeichnung	LINK
1	Einbautaster	Amazon / Metzler



HoneyPi

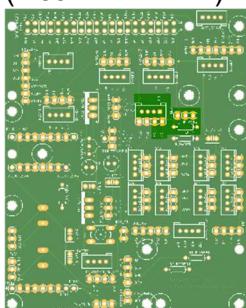
Menge	Bezeichnung	LINK
1	Taster für die Platine	Aliexpress
1	Kondensator, Keramik, 22nF, >= 5V	
1	Widerstand 50 KOhm ($\leq 1\%$)	

7. 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 130Ω . Damit Du nicht rechnen musst, habe ich eine Tabelle mit Standardwerten zur GPIO Spannung von 3,3v erstellt:

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

*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 Rpi GPIOs fließt nicht zu hoch sein darf. Meine persönliche Grenze ist bei 30mA, wissend, das vielleicht etwas mehr geht.

Bei einer Betrachtung sind alle 3,3V Verbraucher am Rpi, die zur gleichen Zeit aktiv sind, zu beachten!

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



HoneyPi

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

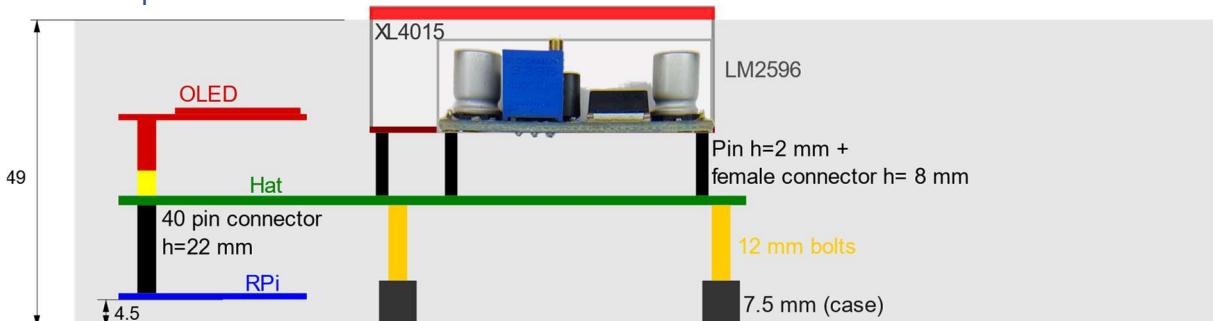
8. Gehäuse

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 RPi-HAT-LM2596



Das Gehäuse ist nicht für das Aufstecken des WittiPy geeignet.

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

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

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

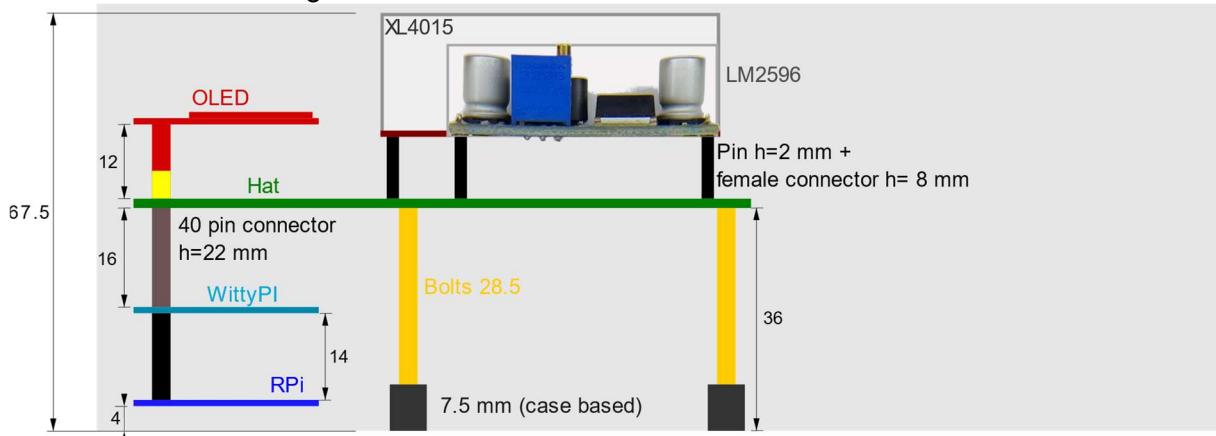
Alternativ können auch Stehbolzen oder Muttern, z.B. mit 2K Klebstoff eingeklebt werden.

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



HoneyPi

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/B00KJLWV1U)

🐝 Kabel für die Spannungsversorgung

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

→ Für die Auslegung gibt es Rechner im Internet.

🐝 Datenleitungen

HX711-Platine 4x $\geq 0,34 \text{ mm}^2$

HX711-Wägezelle: 4x $\geq 0,34 \text{ mm}^2$ geschirmt

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.

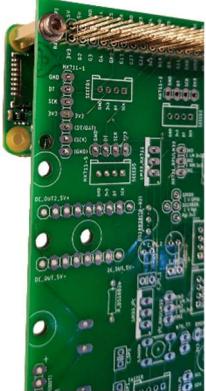
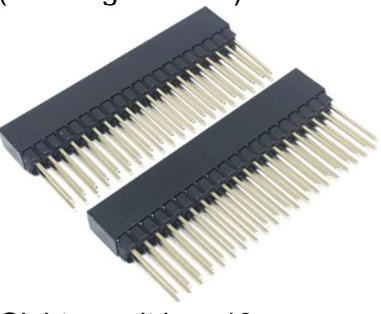
Die max. Länge kann über verschiedenen Rechner im Internet ermittelt werden



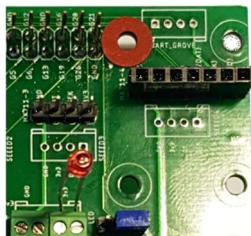
HoneyPi

9. Raspberry Pi Anschluss (40 pol.)

40 Pins für den RPI

	Stacking Header (Bild zeigt 2 Stück)			
Giebt es mit $h = \sim 19 \text{ mm}$ oder $h = \sim 22 \text{ mm}$				

⚠ 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 M2,5x14lg.

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

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

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

10. Gewichtsmessung

 **HX711 A/D-Wandler für die Wägezellen**

Die Platine ist für den direkten Aufbau von zwei verschiedene Typen von A/D-Wandlern 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 3 HX711 Anschlüsse über Steckverbindungen angeschlossen werden. Somit sind 5 HX711 anschliessbar. Bei einem direkten Aufbau muss für eine Abstützung gesorgt werden.

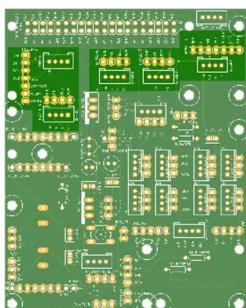


HoneyPi

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 „Honey-PI Sensor Board“ geführt.

⚠ HX711 Modulübersicht

Typ „Grün“	Typ „Violett“	Typ „Red/Sparkfun“	Typ „Red“
A green printed circuit board with a central IC and various pins labeled E-, E+, A-, A+, B-, B+, GND, VCC, SCK, CS, DT, and JTAG. A red circle highlights the GND pin.	A purple printed circuit board with a central IC and various pins labeled RED, BLK, UHT, GRN, YLH, DAT, CLK, and GND. A red circle highlights the DAT pin.	A red printed circuit board with a central IC and various pins labeled RED, BLK, UHT, GRN, YLH, DAT, CLK, and GND. A red circle highlights the DAT pin.	A red printed circuit board with a central IC and various pins labeled UDDI, UCC, DAT, CLK, and GND.
Typ „Grove“		Typ „XFW“	
A blue printed circuit board with a central IC and various pins labeled GND, UCC, DOUT, SPSCK, E-, E+, DAT, CLK, and VCC.	A green printed circuit board with a central IC and various pins labeled GND, DAT, CLK, and VCC. The label "XFW - HX711" is visible at the bottom.	?	?

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.

⚠ 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

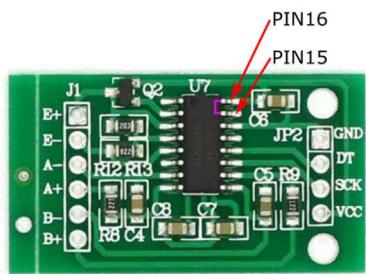


HoneyPi

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“

💡 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. $9\text{g}/^\circ\text{C}$ 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.



HoneyPi

11. Temperatursensor DS18B20 (1Wire)

Hierfür werden ein Widerstand mit $4,7\text{ k}\Omega^*$ 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 höherer Widerstand ($10\text{ k}\Omega$) 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.

	3,3V vom RPi	3,3V vom GPIO09	3,3V vom LF33CV (HX711)
RPi	Per Software schaltbar	RPi unabhängig 3,3V	

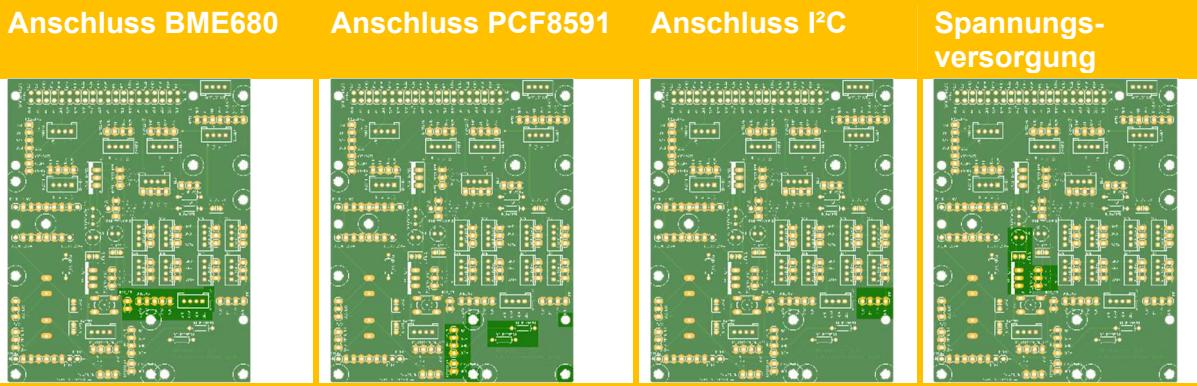
Menge	Bezeichnung	LINK
X	DS18B20 - Edelstahlgehäuse	Aliexpress/ DS18B20 -Edelstahlgehäuse
1	DS18B20 - TO92	Aliexpress/ DS18B20 - TO92
1	Widerstand 4,7 KOhm	



HoneyPi

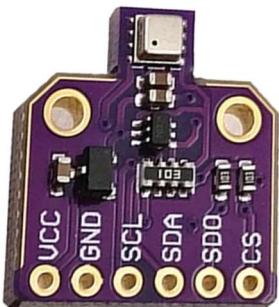
12. I²C Schnittstellen / Sensoren

💡 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.



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*).





HoneyPi

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 3,3v oder GND erreicht werden kann muss ausgemessen werden, bevor man den SD0 anschließt.

Alternative I²C Sensoren (noch nicht in HoneyPi integriert)

ATH10

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

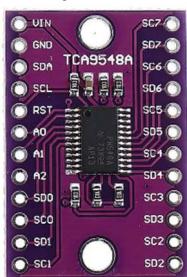
SHT31-D ($\pm 2\%$ & $\pm 0.3\text{ }^\circ\text{C}$)

Einen schönen Vergleich einiger Sensoren findest Du hier:

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

I²C Adressen

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 Honey-Pi Software eine Anpassung erfolgen. Dieser Ansatz wurde verworfen.



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

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

Menge	Bezeichnung	LINK
X	BME680	Aliexpress/ BME680
X	BME280	Aliexpress/ BME280 3,3v
1	Pin Header, männlich	
1	Grovebuchse	



HoneyPi

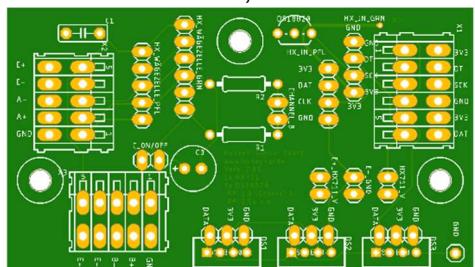
13. Kleinteile

Auflistung ist noch offen.

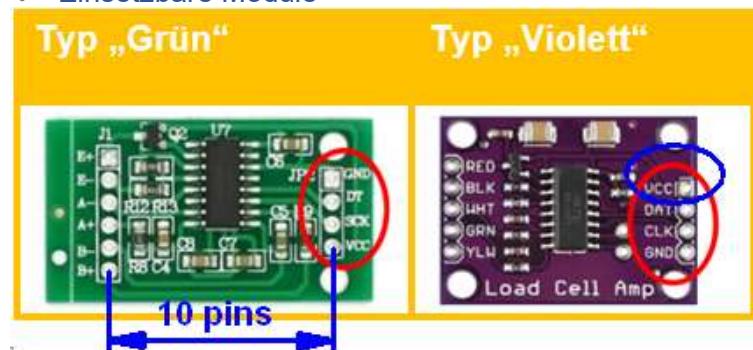
Bitte erstmal in den Schaltplan gucken.

14. 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.



💡 Einsetzbare Module



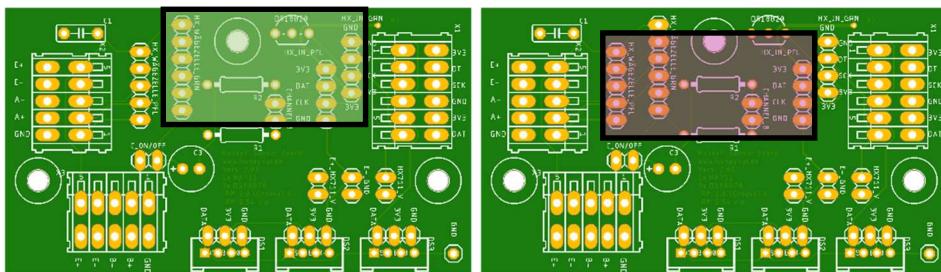
Typ „Grün“ hat einen Pinreihenabstand von 10 Pins ($10 \times 2,54 \text{ mm} = 25,4 \text{ mm}$).

Typ „Violett“ hat einen Pinreihenabstand von 11 Pins ($11 \times 2,54 \text{ mm} = 27,94 \text{ mm}$).

Auf die Pinbelegung ist zu achten!



HoneyPi



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.

15. Trouble shooting

💡 HX711

💡 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



HoneyPi

```
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: -- - - - - - - - - - - - - - - - - - - - - -
```

⚠ I²C Bus: Verbindungsausfälle, o.ä. Probleme:

Der I²C Bus ist so spezifiziert, das die Kapazität der Busleitung nicht über 400pF (besser nur 200pF) 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, die Übertragungsfrequenz zu reduzieren und das soll ca. 30m 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=3v3 oder low=0v 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 KOhm (GPIO 2 & 3: 1,8 k Ohm).

Beim I²C-Bus können alle Geräte den Bus auf GND ziehen. Keines der Gerät am I²C Bus kann einen High-Pegel (auf SDA oder SCL) erzeugen. 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



HoneyPi

jetzt mehrere dieser Module zusammen, so sind alle diese Pullups parallel geschaltet. Hat man mehrere solcher Module, kann der resultierende Widerstand so niedrig werden, dass die Ausgänge überlastet werden könnten. Die Lösung ist dann denkbar einfach: Die Pullups müssen auf einigen Modulen deaktivieren werden.

→ mindestens auf einem muss er bleiben.

Die GPIO 2 und 3 (für den I²C) nur 1,8 KOhm Pullupwiderstände.
Viele Module haben 10 KOhm, einige 4,7 KOhm Pullupwiderstände.

Beispiel Rechnung 1:

(R1=1,8KOhm; R2 =10 KOhm)

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

$$R_{ges.} = 1,8 \text{ KOhm} \times 10 \text{ KOhm} / (1,8 \text{ KOhm} + 10 \text{ KOhm})$$

$$R_{ges.} = 18 \text{ KOhm} / (11,8 \text{ KOhm})$$

$$R_{ges.} = 1,52 \text{ KOhm}$$

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

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

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

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

Beispiel Rechnung 2:

(R1=1,8KOhm; R2 =4,7 KOhm)

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

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

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

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

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

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

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

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

⚠ I²C Bus:Übertragungsfrequenz reduzieren:

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
```

16. 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!



HoneyPi

17. Tabelle Verbraucher

Pos.	Menge	Bezeichnung	Spannung [V]	Strom[mA] (Menge =1)	Strom[mA] (Menge =n)	Am Raspi (3v3)
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	2V (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		
7	1	ATH10	3,3			
8	1	SHT31	3,3			
9	1	HDC1080	3,3			
10	1	LCD	5V			
11	1	WittyPi	5V	7		
12	1	PCF8591	3,3	2,81		
13	1	USB / UMTS Modem Huawei E303	5V	750 mA (Peak?)		
14		USB / UMTS Modem Huawei E3531i-2	5V	700mA (Peak?)		
15	1	LM2596	12V	5-10mA		
16		XM4015		max. 30mA		
17		LF33CF				
18						

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

18. Beispiel für eine Stromüberschlagsrechnung

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

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

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

$$P_1 = 3,3 \text{ V} \times 0,15 \text{ A}$$

$$P_1 = 0,495 \text{ W}$$

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

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

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

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



HoneyPi

Summe des 2. Stromkreises (5 V):

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

$$P_2 = U \times I_{5V}$$

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

$$P_2 = 4,6 W$$

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:

$$P_{Ges.} = P_1 + P_2$$

$$P_{Ges.} = 0,04125 A + 0,3833 A$$

$$\underline{P_{Ges.} = 0,4125 A (@12V)}$$

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

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

Belegung der GPIOs über den 40 Pol. Steckverbinder zur Platine							
Unbelegt	3V Power	1	2	5V Power	Power in		
I2C-BUS	GPIO2 SDA1 I2C	3	4	5V Power	Unbelegt		
I2C-BUS	GPIO3 SCL1 I2C	5	6	Ground	Unbelegt		
WittyPi	GPIO4	7	8	GPIO14 UART0_TXD	RX - UMTS Modem		
GND	Ground	9	10	GPIO15 UART0_RXD	TX - UMTS Modem		
WittyPi	GPIO17	11	12	GPIO18 PCM_CLK	Unbelegt		
HX711_DT_(3)	GPIO27	13	14	Ground	Unbelegt		
HX711_SCK_(3)	GPIO22	15	16	GPIO23	HX711_DT_(4)		
3,3V (Taster, diverse)	3,3V Power	17	18	GPIO24	HX711_SCK_(4)		
Unbelegt	GPIO10	19	20	Ground	Unbelegt		
1-Wire DS18b20 3,3V	GPIO9	21	22	GPIO25	HX711_DT_(5)		
1-Wire, DS18b20 Data	GPIO11	23	24	GPIO8 SPI0_CE0_N	Unbelegt		
Unbelegt	Ground	25	26	GPIO7 SPI0_CE1_N	Unbelegt		
RESERVED	I2C ID EEPROM	27	28	I2C ID EEPROM	RESERVED		
HX711_DT_(1)	GPIO5	29	30	Ground	Unbelegt		
HX711_SCK_(1)	GPIO6	31	32	GPIO12	HX711_DT_(2)		
HX711_SCK_(2)	GPIO13	33	34	Ground	Unbelegt		
Unbelegt	GPIO19	35	36	GPIO16	Taster		
HX711_SCK_(5)	GPIO26	37	38	GPIO20	Unbelegt		
Unbelegt	Ground	39	40	GPIO21	LED		



HoneyPi

20. 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

21. 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 Honey-Pi.de selbst: <https://www.honey-pi.de/Einkaufsliste/>



HoneyPi

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

22. 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

23. 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 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 Tage. Hast Du mehr Zeit und möchtest Geld sparen, dann bietet sich eine Bestellung 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.

Bedarfsanalyse

Funktionsliste

Materialliste

Beschaffung

Aufbau

💡 Bauvorschlag I

Bedarfsanalyse und Funktionsliste:

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

Stromversorgung: 12V, mit Spannungsregler LM2596 (steckbar)

Materialliste und Beschaffung:

💡 Aufbau:

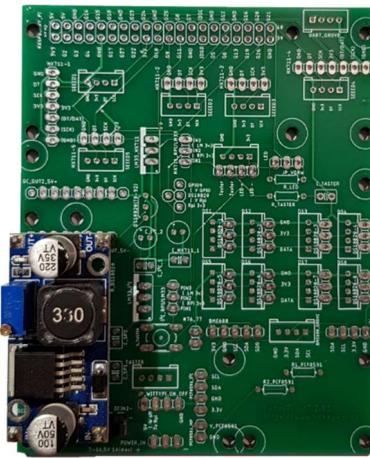
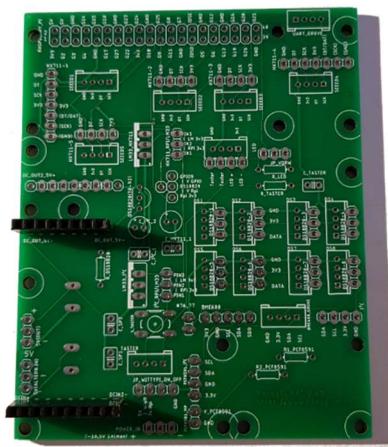
Zum Aufbau benötigst Du eine Buchsenleiste, die Du zurechtsägen oder einfach mit dem Seitenschneider zurechtschneiden kannst. Da ich optional das größere Modul XL4015, anstatt des LM2596 aufbauen möchte, kommt unten die 9 pol. Buchsenleiste und oben eine 8 und 9 pol. Buchsenleiste hin.

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



HoneyPi

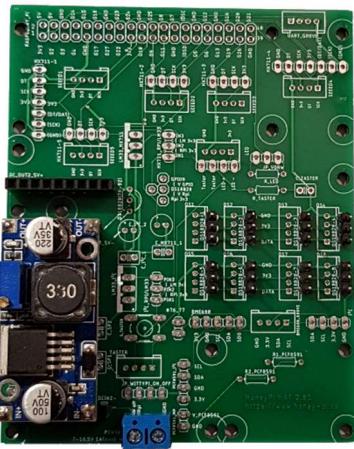
LM2596



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.

12V in



Jumper RPi / WittyPi:

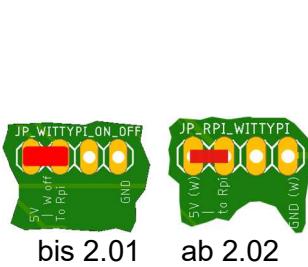
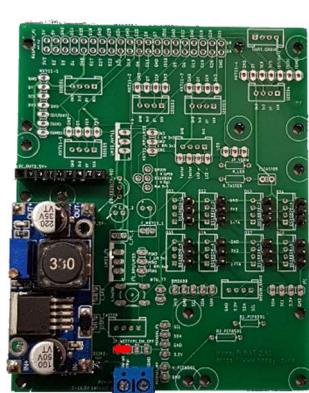
Ohne WittyPi muss hier gebrückt werden. Mit Pins und Jumper oder Draht und Lötzinn.
Alternativ kann man hier auch den WittyPi dazwischen klemmen.



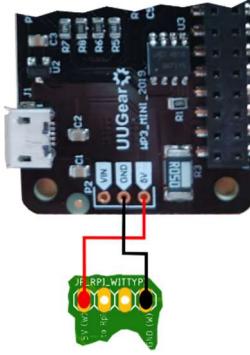
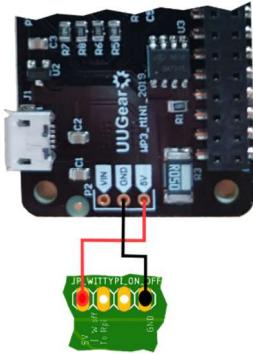
HoneyPi

Jumper RPI / WittyPi

Jumper / Brücke Stromversorgung ohne WittyPi



Anschluss WittyPi (Jumper entfällt, vereinfachte Darstellung*)



Beim Einsatz des WittyPi wird die Brücke nicht gesetzt. Hier werden nur die 5V, 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:

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

Ab Version 2.02:

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

*Der WittyPi muss natürlich auch auf den 40pol. Steckverbinder, zwischen Raspberry und Platine, gesteckt werden.

Diese Anleitung bezieht sich auf den WittyPi 3 mini.

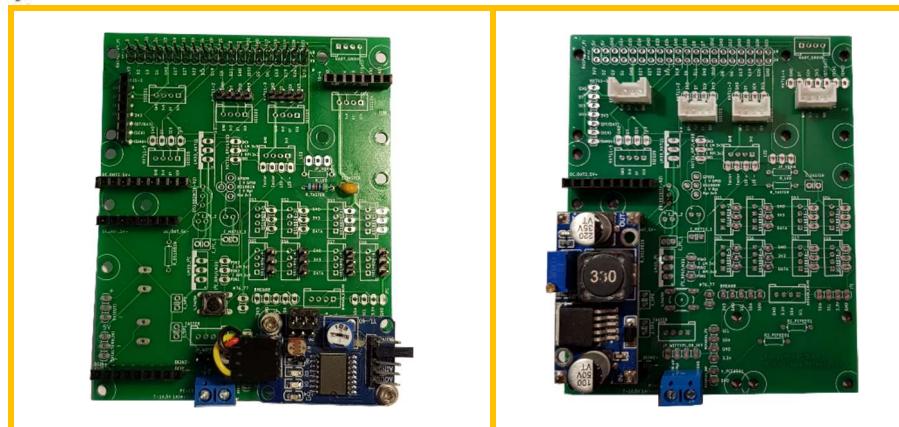
HX711:

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

HX711

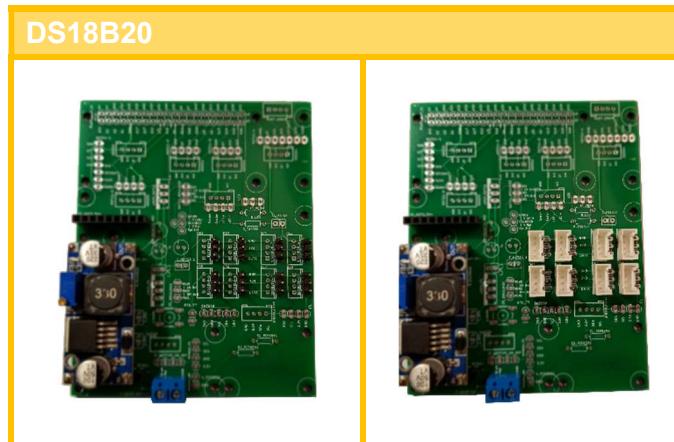


HoneyPi



DS18B20

2,54mm 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.



RPi 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.

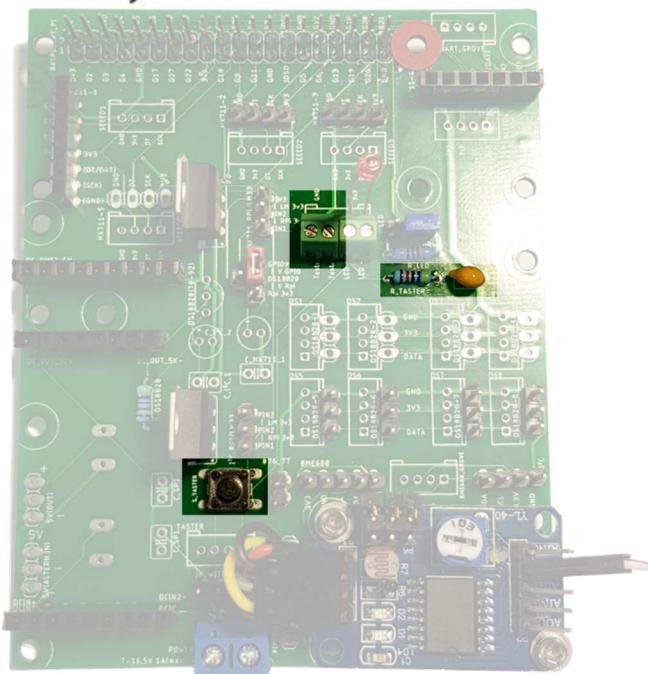
- Wenn man diese Platine auf einen RPi 3 oder 4 aufsetzen möchte und die Bauhöhe und Orientierung keine Rolle spielt, steckt man ein 2. Stacking Header als Höhenadapter auf und kommt so über die USB & LAN Anschlüsse des RPi.

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 KOhm 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 Ohm (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 RPI

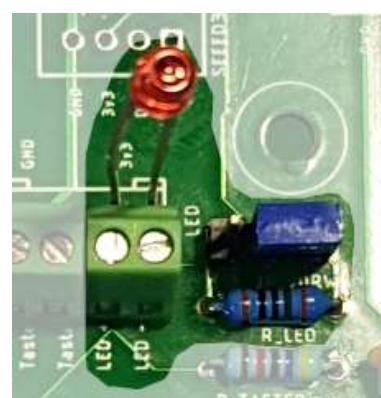
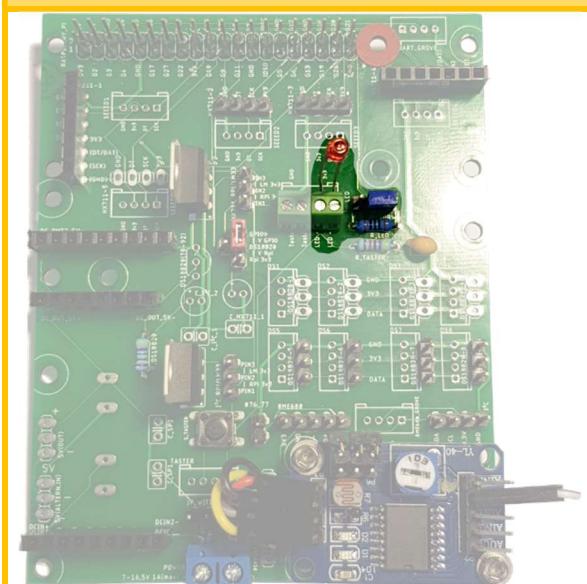
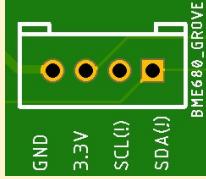
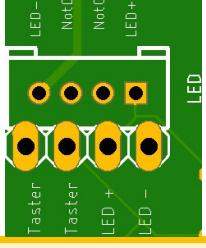


Bild zeigt noch die alte Version.
Jumper ist nun nur noch zum
deaktivieren des LED-
Vorwiderstandes notwendig.



HoneyPi

24. Änderungshistorie Platine

Version	Datum	Änderung
P2.00	01.11.2020	Erster Wurf, 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 NoC (not Connected) bei der LED-Grovebuchse hinzugefügt. 

25. Änderungshistorie Dokument

Version	Datum	Änderung
D2.01	27.09.2020	Erster Wurf
D2.02	20.10.2020	HX711 „red“ ergänzt. HX711 Sensor Board HX711 + DS18b20
D2.02	06.11.2020	<p>Änderungen im Bereich BME680/280 zu SD0 Sensor Board HX711 und DS18b20 hinzugefügt. Änderungshistorie: Versionsnummern um D (für Dokument) und P (für Platine) erweitern.</p>
D2.03		Einige Teilelisten mit Links eingefügt.



HoneyPi

		Belegungsliste 40pol. hinzu. GPIO Belegungsliste hinzu Taster Entprellen korrigiert. Hinweise zur Kabellänge, EMV und zur Wägezelle hinzugefügt.