



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 RPI 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 eine nach und nach Erweiterung bedacht 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. Kauft 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.

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 mit einem Hinweis auf meine Arbeit würde mich sehr freuen.

*Gerne erstelle ich, mit Dir zusammen, später auch ein Layout für den Raspberry Pi 3/4.



Inhalt

1.	PLATINE V2.01.....	4
2.	FUNKTIONSLISTE DES BOARDS:	4
	• <i>Spannungsversorgung und Messung</i>	4
	• <i>Taster (Wartungsmodus=an/aus)</i>	4
	• <i>LED</i>	4
	(blinkt 3x, wenn Bootvorgang abgeschlossen und leuchtet im Wartungsmodus)	4
	• <i>Gewichtsmessung</i>	5
	• <i>Temperaturmessung</i>	5
	• <i>Diverse I²C Bus-Sensoren</i>	5
	• <i>WittiPi</i>	5
	• <i>Timer Board</i>	5
3.	SPANNUNGSVERSORGUNG	6
	• <i>Übersicht</i>	6
	• <i>LM2596 Modul</i>	6
	• <i>XL4015 Modul</i>	7
	• <i>5 V</i>	7
	• <i>Festspannungsregler LF33CV für 3,3 V</i>	7
	• <i>Brücken (Jumper) HX711</i>	8
	• <i>Brücken (Jumper) I²C</i>	8
	• <i>Brücken (Jumper) DS18B20</i>	8
4.	ABSICHERUNG	9
5.	SPANNUNGSMESSUNG	10
	• <i>PCF8591 (I²C)</i>	10
	• <i>Spannungsteiler</i>	10
	• <i>Musterberechnungen: Spannungsteiler</i>	11
6.	TASTER	12
	• <i>Taster entprellen</i>	12
	• <i>Musterberechnungen: Entprellen des Tasters mit Kondensator</i>	12
7.	LED	13
	• <i>Vorwiderstand für die LED</i>	13
8.	GEHÄUSE	14
	• <i>Übersicht</i>	14
	• <i>Kabel & Stecker</i>	14
	• <i>Kabel für die Spannungsversorgung</i>	14
	• <i>Datenleitungen</i>	14
9.	RASPBERRY PI ANSCHLUSS (40 POL.).....	15
10.	GEWICHTSMESSUNG	15
	• <i>HX711 A/D-Wandler für die Wägezellen</i>	15
	• <i>Anschluss, Kabel und Abschirmung</i>	16
	• <i>HX711 Modulübersicht</i>	16
	• <i>Typ „Grün“ analog GND(E-) und digital GND verbinden</i>	16
	• <i>Typ „Grün“ Sampling-Rate erhöhen</i>	16
	• <i>Typ „Violett“</i>	16
	• <i>Typ „Red Sparkfun“</i>	17
	• <i>Typ „Red“</i>	17
	• <i>Typ „Grove“</i>	17



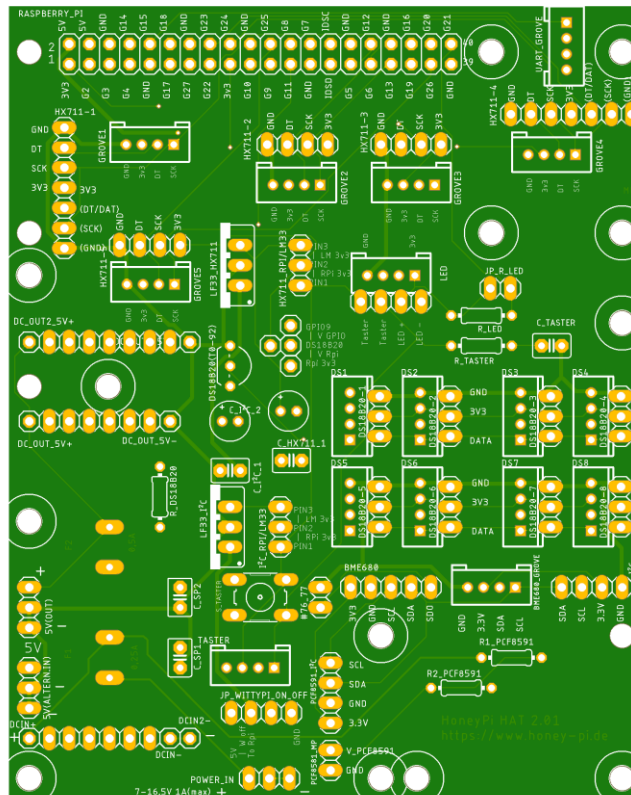
HoneyPi

•	Typ „XFW“.....	17
•	Wägezelle	17
11.	TEMPERATURSENSOR DS18B20 (1WIRE)	18
12.	I ² C SCHNITTSTELLEN / SENSOREN	19
•	Übersicht.....	19
•	BME680	19
•	BME280	19
•	Alternative I ² C Sensoren (nicht in HoneyPi integriert)	20
•	I ² C Adressen	20
13.	TROUBLE SHOOTING.....	21
•	HX711	21
•	Generelles zum I ² C Bus (Links):	21
•	I ² C Bus Teilnehmer prüfen:	21
•	I ² C Bus: Verbindungsausfälle o.ä. Probleme:.....	22
•	I ² C Bus:Übertragungsfrequenz reduzieren:.....	22
14.	SOFTWARE KONFIGURATION.....	22
15.	TABELLE VERBRAUCHER	22
16.	TEILEBESCHAFFUNG UND EINKAUFLISTE	23
17.	AUSBlick	23
18.	BAUANLEITUNG(EN).....	24
•	Vorbereitung oder „Wie fange ich an“?	24
•	Bauvorschlag I	24
19.	ÄNDERUNGSHISTORIE PLATINE	28
20.	ÄNDERUNGSHISTORIE DOKUMENT	28



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://ilcpcb.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 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.

Gewichtsmessung

3+2x HX711 (2x könnten aufs Board gelötet/gesteckt werden, wobei das nicht empfohlen wird).

Aufgelötet werden können 2x Grüne oder 2x violette Modul

3x Schnittstelle zum Sensor Board 2.x

Temperaturmessung

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)

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 Honey-Pi Unterstützung wurde noch nicht angegangen



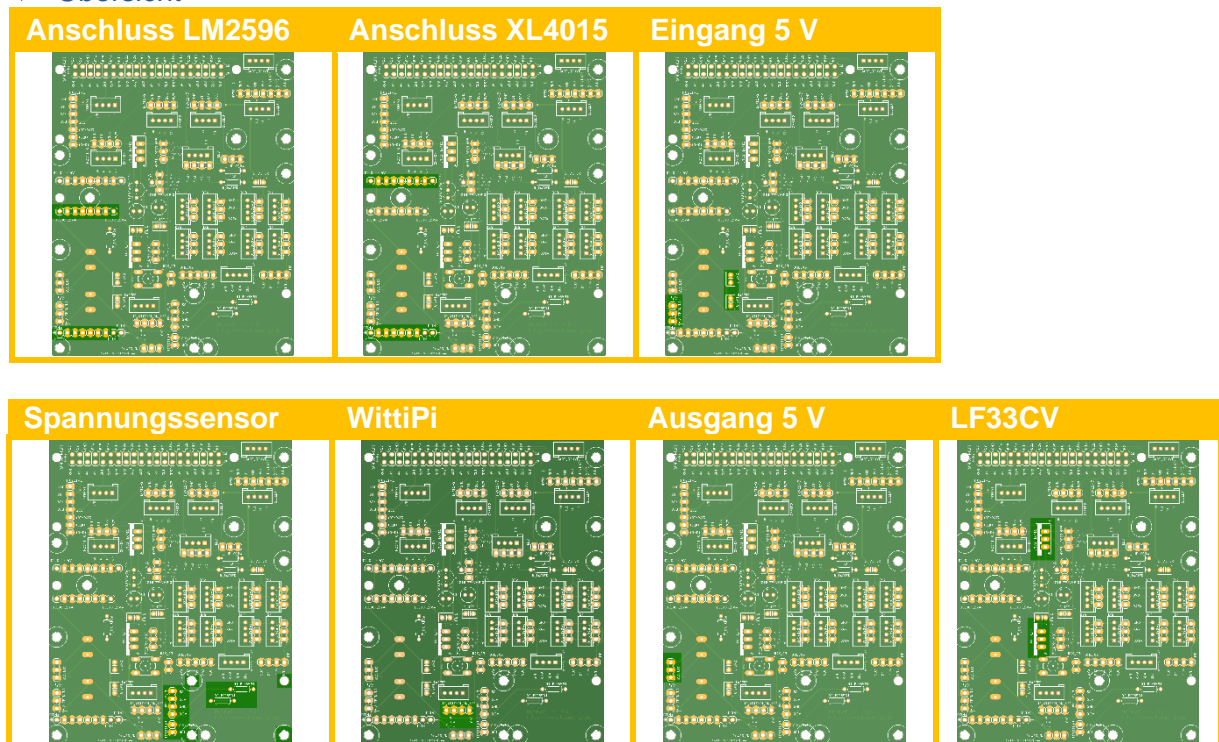
HoneyPi

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.

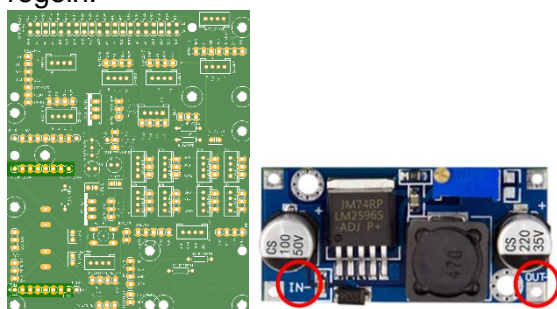
Es können hier die größeren RM 5,08 mm Klemmsteine und alternativ auch die kleineren 2,54 mm Klemmsteine eingesetzt werden. Dazu sind von den 3 Pin-Holes ein Pin-Hole für + und 2 für GND vorgesehen. Der kleine Klemmstein kommt in die Pin-Holes 1 und 2 und der große Klemmstein in die Pin-Holes 1 und 3.

Übersicht



LM2596 Modul

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!



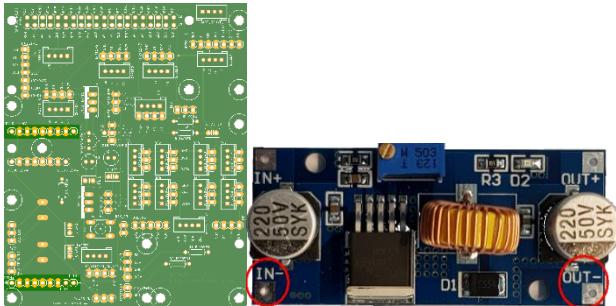
HoneyPi

- ⚠ 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

Alternativ z.B. für eine Solarversorgung.

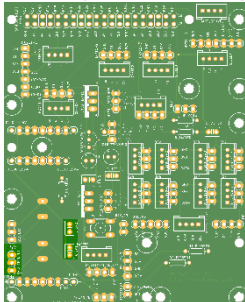
Im Heise.de Forum habe ich von „spontanen“ ausfällen des LM2596 gelesen. Daher habe ich dieses alternative Modul vorgesehen. Auch wenn weitere Verbraucher geplant werden sollten, ist dieses Modul eine Alternative, da ein wesentlich höherer Strom bereitgestellt werden kann.



- ⚠ Beim Aufbau ist der Hinweis in-/out- (roter Kreis) zu beachten und die Ausgangsspannung von 5 V muss vorab 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.

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.

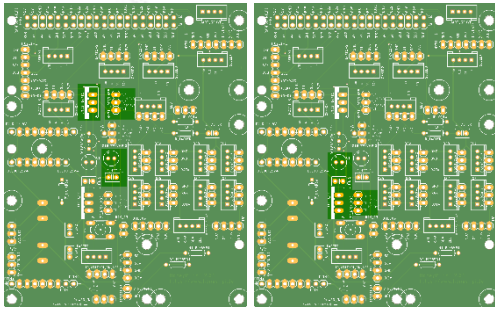


Festspannungsregler LF33CV für 3,3 V

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. Grösser als 100 µF mach keinen Sinn, bzw. kann sogar die Regelung des LF33CV verlangsamen.



HoneyPi



Brücken (Jumper) HX711

HX711 versorgt über RPi	HX711 versorgt über LF33cf	HX711 & DS18B20 werden über LF33cf versorgt

Brücken (Jumper) I2C

I2C Busteilnehmer versorgt über RPi	I2C Busteilnehmer versorgt über LF33cf

Brücken (Jumper) DS18B20


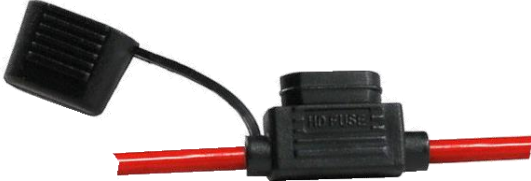
DS18B20 versorgt über RPi (3,3V)	DS18B20 versorgt über RPi (GPIO09)	DS18B20 versorgt über RPi



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 Fotovoltaikanlage 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 dem 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 auch 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 DU machst Dir Gedanken zu einer Absicherung Deiner Anlage!

Leitungssicherung (Hauselektrik)	Leitungssicherung (KFZ-Bereich)
	

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

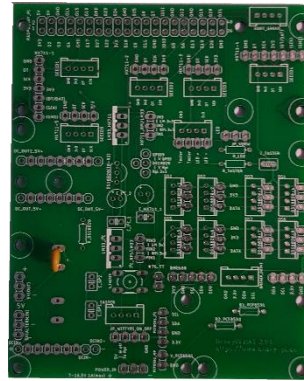
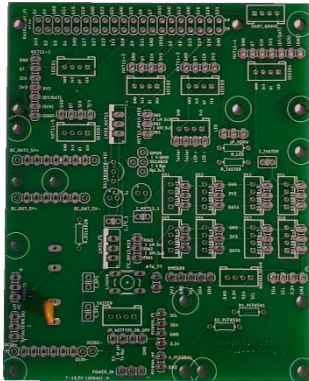
Zur Absicherung der Platine sind zwei Möglichkeiten zum Anschluss einer Sicherung vorhanden.

1x für die Eingangsspannung vor den Spannungsreglern (LM2596 o. XM4015) und 1x bei 5V Spannungsversorgung.

Vor dem Spannungswandler	Nach dem Anschluss 5V
--------------------------	-----------------------



HoneyPi



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.

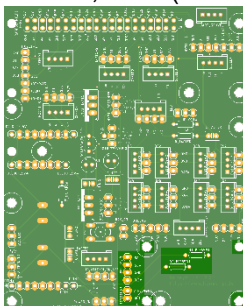
5. Spannungsmessung



PCF8591 (I²C)

Der PCF8591 kann über 3 Stehbolzen (M2,5, Lg.7mm) 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 k Ω + 7,5 k Ω (oder deren Vielfache).



Spannungsteiler

Das Verhältnis 4:1 sollte für eine 12V 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 k Ω / 7,5 k Ω oder 40 k Ω / 10 k Ω 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 werde.

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



HoneyPi

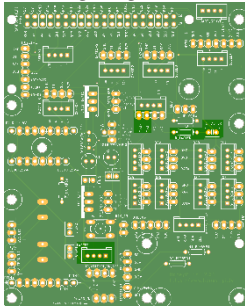
 Musterberechnungen: Spannungsteiler

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



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 entprellen

Zum Entprellen können ein Kondensator und ein Widerstand aufgelötet werden. Solltest Du einen, von der Materialempfehlung abweichenden Taster einsetzen wollen, so kannst Du den Kondensator und Widerstand entsprechend anders auswählen.

Musterberechnungen: Entprellen des Tasters mit Kondensator.

Default (OnBoard Taster)

RC-Konstante= $R * C$

$t = 47 \text{ kOhm} * 100 \text{ nF} = 47.000 \text{ Ohm} * 0,000.000.10 \text{ F}$

$t = 4,7 \text{ ms}$

Alternative 1

RC-Konstante= $R * C$

$t = 50 \text{ kOhm} * 47 \text{ nF} = 50.000 \text{ Ohm} * 0,000.000.047 \text{ F}$

$t = 2.35 \text{ ms}$

Alternativ 2

RC-Konstante= $R * C$

$t = 50 \text{ kOhm} * 22 \text{ nF} = 50.000 \text{ Ohm} * 0,000.000.022 \text{ F}$

$t = 1.1 \text{ ms}$

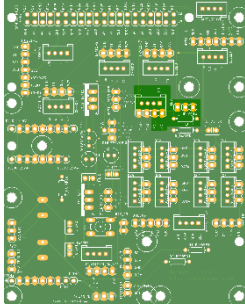


HoneyPi

7. LED

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

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



Vorwiderstand für die LED

Der Vorwiderstand soll den Strom begrenzen, also verhindern, dass die „LED“ durchbrennt und dass 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 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 Rpi GPIOs fließt nicht zu hoch sein darf. Meine persönliche Grenze ist bei 30mA, wissend, dass vielleicht etwas mehr geht. Bei einer Betrachtung sind alle 3,3V Verbraucher, die zur gleichen Zeit aktiv sind, zu betrachten!



HoneyPi

8. Gehäuse

Als Gehäuse habe ich mich für ein einfaches Kunststoffgehäuse 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 Werkstoffeigenschaften	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 €

Kabel & Stecker

Ich selbst bevorzuge PG-Verschraubungen. Zwar ist in einem 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

Kabel für die Spannungsversorgung

Nach Auslegung 0,5 mm²-2 mm²

Datenleitungen

HX711-Platine 4x 0,32mm²

HX711-Wägezelle: 4x 0,32mm² geschirmt

Ich verwende Sensorleitungen, die aus einem anderen Projekt über geblieben sind.

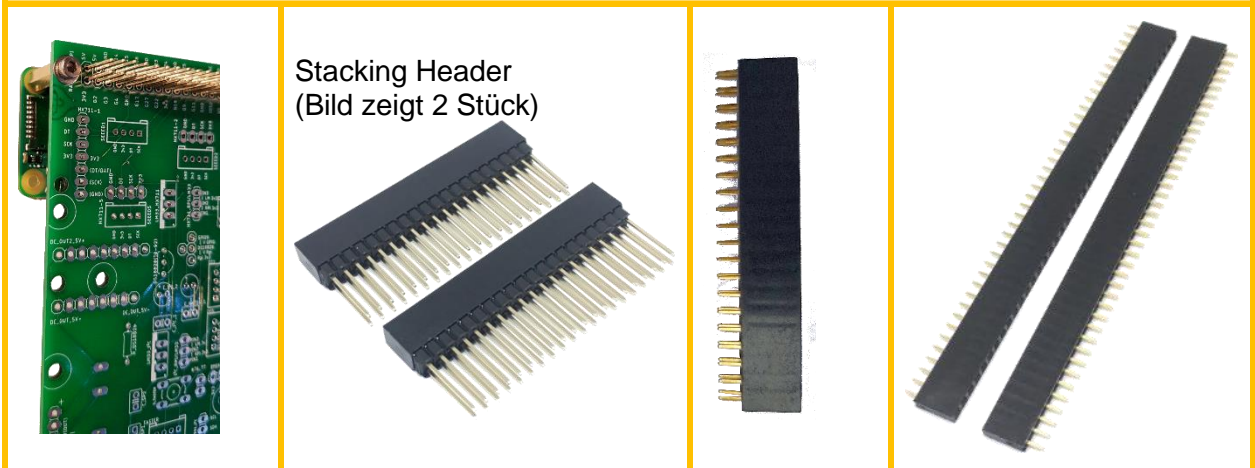
CAT.5E, 6 oder 7 sind sehr günstig und gut verfügbar.



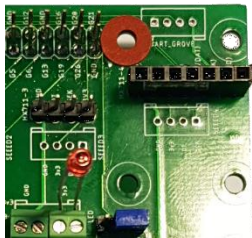
HoneyPi

9. Raspberry Pi Anschluss (40 pol.)

40 Pins für den RPI



⚠ 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)

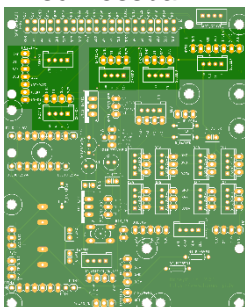
Stehbolzen verwende ich M2,5x14lg.

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

10. Gewichtsmessung

 HX711 A/D-Wandler für die Wägezellen

Die Platine ist für den direkten Aufbau von 2 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.



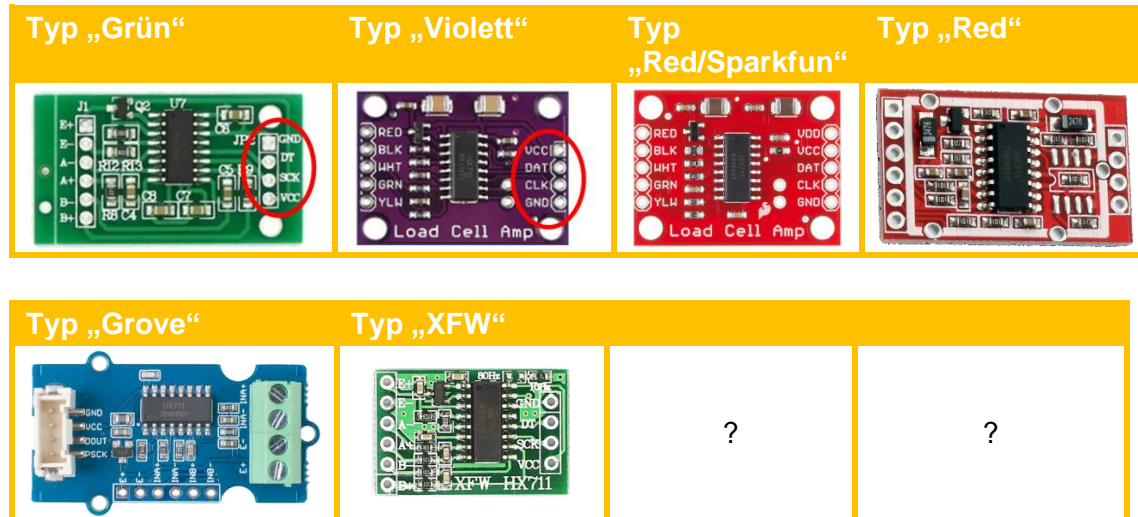


HoneyPi

🔌 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

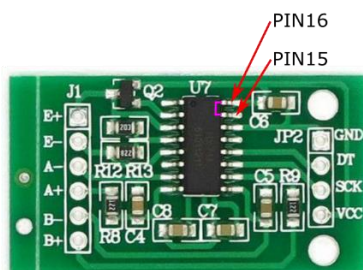


🔌 Typ „Grün“ analog GND(E-) und digital GND verbinden.

Im Netz gibt es verschiedene Meinungen dazu. Wenn der Widerstand zwischen analog GND(E-) und digital GND nicht gegen Null geht und es Kurz- oder Langzeitabweichungen oder auch ggf. auch temperaturabhängige Abweichungen kommt, 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 aber das ist nicht meine Erfahrung. Bestellt, bisher nicht eingetroffen.



HoneyPi

 [Typ „Red Sparkfun“](#)

 [Typ „Red“](#)

Bestellt, bisher nicht eingetroffen.

 [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 nur die Bosche H30A

Abschirmung gut auflegen (Metallgehäuse für den HX711 + IRIS-Verschraubung)

Infos findest Du auf Honey-Pi.de



HoneyPi

11. Temperatursensor DS18B20 (1Wire)

Hierfür werden ein Widerstand mit 4,7 k Ω * 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.

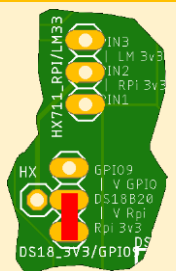
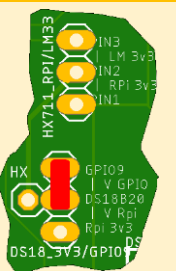
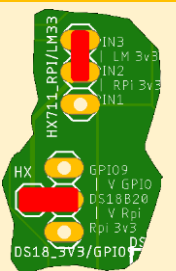
DS18B20 -Edelstahlgehäuse:	
DS18B20 - TO92:	

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

*Bei einer grösseren Menge und auch in Abhängigkeit der Topologie kann eine höherer Widerstand (10 k Ω) 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.

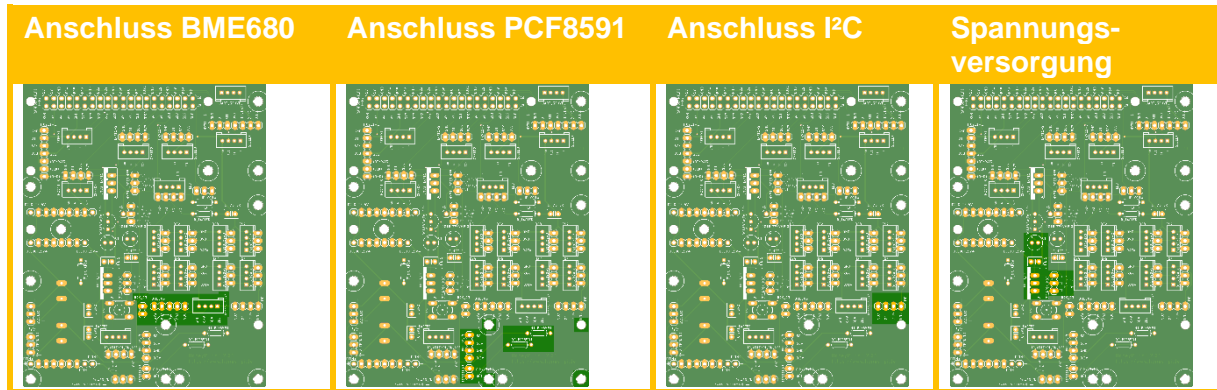
3,3V vom RPi	3,3V vom GPIO09	3,3V vom LF33CV (HX711)
		
RPi	Per Software schaltbar	RPi unabhängige 3,3V



HoneyPi

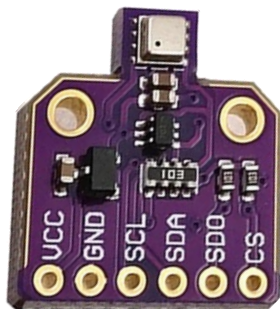
12.I²C Schnittstellen / Sensoren

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 und 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.



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, re), wo die Adressänderung, anders als beim BM680 nicht funktionierte.

Alternative I²C Sensoren (nicht in HoneyPi integriert)

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

HTU21D-F ($\pm 2\%$ & $\pm 1\text{ }^{\circ}\text{C}$)

Si7021 ($\pm 3\%$ & $\pm 0.4\text{ }^{\circ}\text{C}$)

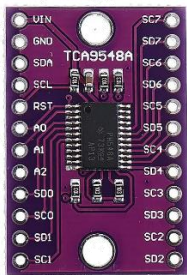
SHT31-D ($\pm 2\%$ & $\pm 0.3\text{ }^{\circ}\text{C}$)

Schöner Vergleich der Sensoren findest Du hier:

http://www.kandsmith.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 Honey-Pi eine Anpassung erfolgen.



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

➔ Siehe „I²C Bus: Verbindungsausfälle o.ä. Probleme“



HoneyPi

13. 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

```
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:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@HoneyPi:~$
```



HoneyPi

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.

I2C Bus:Übertragungsfrequenz reduzieren:

In die config.txt einfügen:

```
dtoverlay=i2c-arms=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
```

14. 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.

15. 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		HX711	3,3	1,5		
3	1	LED Typ: Rot, 2V, 10mA, Vorwiderstand Rv=130 Ohm	2V (an 3v3)	10		
4		DS18B20	3v3	1,5		
5		BME280	3v3	0,05		
6		BME680	3v3	12		
7		ADS1234	3v3			
8		WittyPi	5V	7		
9		PCF8591	3v3	2,81		
10		USB / UMTS				



HoneyPi

16. 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 lieb und hat eine Einkaufsliste bei Reichelt erstellt:

- HoneyPi Platine 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/>

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

17. 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

18. 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://jlcpcb.com/> dauert ca. 10 Tage. Hast Du mehr Zeit und möchtest Geld sparen, dann bietet sich eine Bestellung bei Aliexpress.com, Bangood.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:

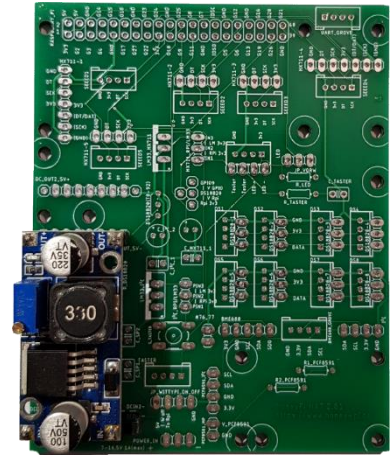
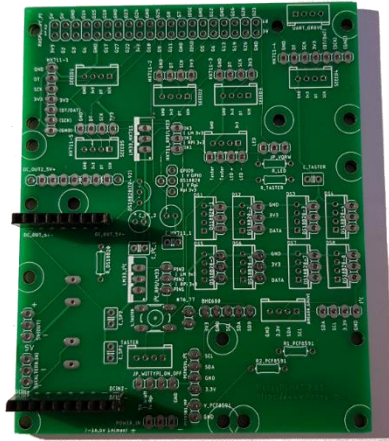
.....

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 pol. Buchsenleiste hin.

- ➔ Beim Zerschneiden mit 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.

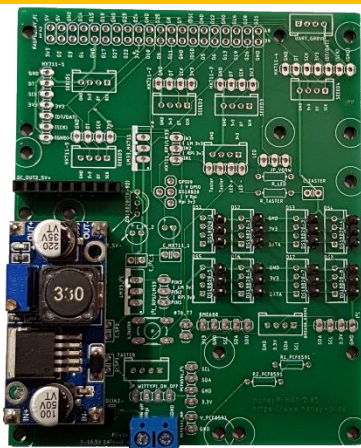
LM2596



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 Klemmblock gesetzt werden.

➔ Von den 3 Pins sind 2 mit GND verbunden.

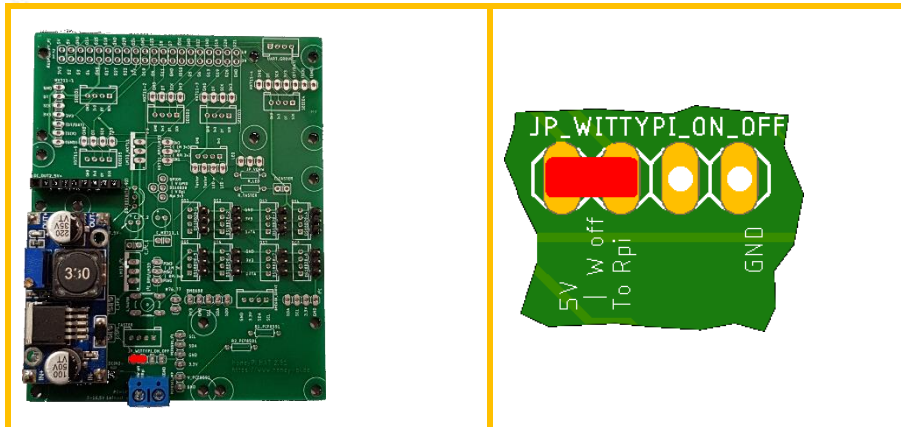
12V in



Jumper RPI / WittyPi:

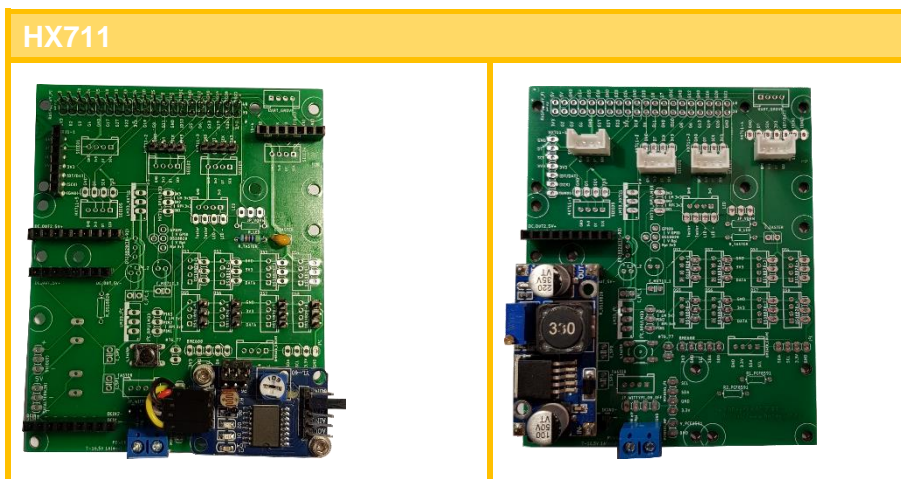
Ohne WittyPi muss hier gebrückt werden. Mit Pins und Jumper oder Draht und Lötzinn Alternativ kann man hier auch den WittyPi dazwischen klemmen.

Jumper RPI



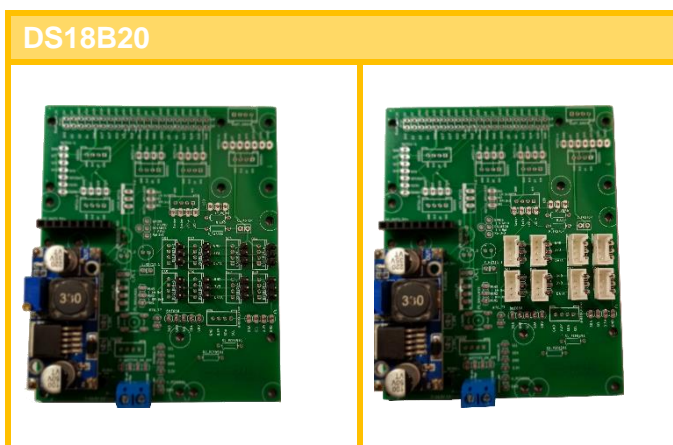
HX711:

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



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.





HoneyPi

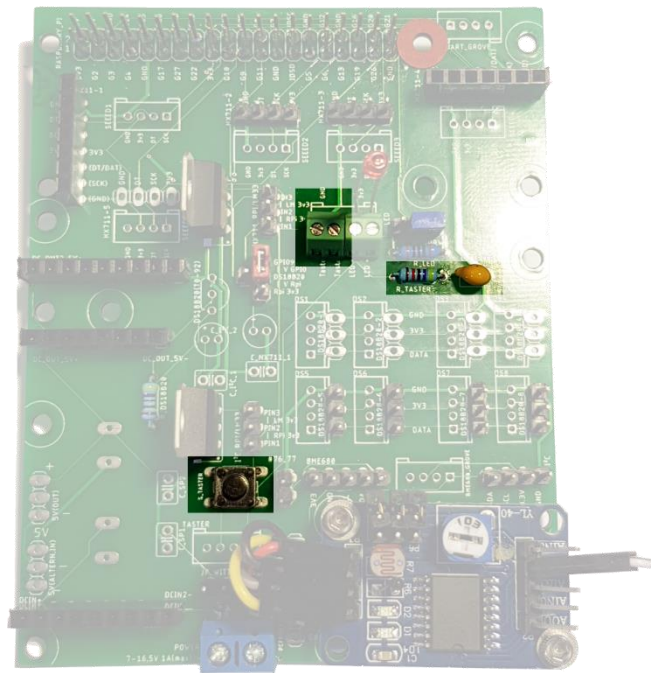
RPi Anschluss

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, steck 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.



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.

*In der Version 2.01 ist der Jumper nur notwendig, wenn der Widerstand gebrückt werden soll.

40 Pins für den RPi



HoneyPi

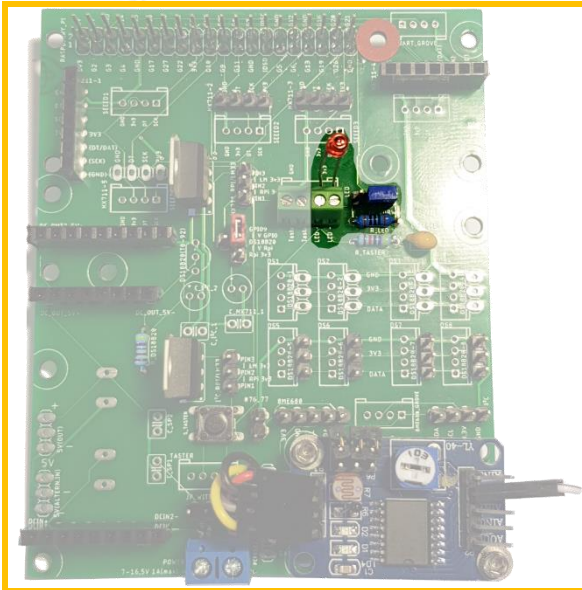


Bild zeigt noch die alte Version.
Jumper ist nun nur noch zum
deaktivieren des LED-
Vorwiderstandes notwendig.

19. Änderungshistorie Platine

Version	Datum	Änderung
2.00	01.11.2020	Erster Wurf, nicht öffentlich.
2.01	27.09.2020	<ul style="list-style-type: none"> - Diverse kleine Fehlerkorrekturen. - LED Jumper vereinfacht. - Grove-Anschluss UART 90° gedreht.

20. Änderungshistorie Dokument

Version	Datum	Änderung
2.01	27.09.2020	