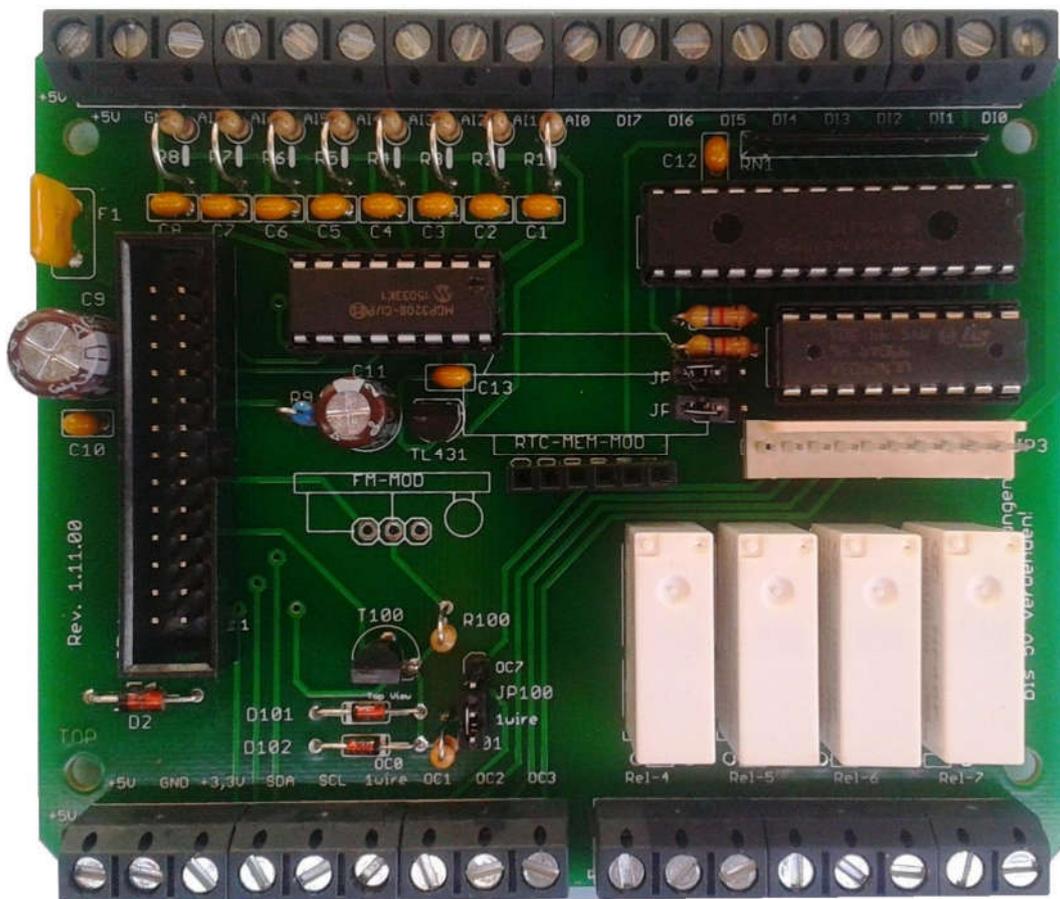


Hubo – das universelle Hutschienenboard

Technische Dokumentation, Sicherheitshinweise, Aufbau, Inbetriebnahme, Optionen und mehr...



Stand vom 24.11.2019, Revision 1.10

FTL – Auerbach, Inhaber Dag Auerbach
St.-Vitus-Str. 41, 82205 Gilching

Copyright 2019 FTL – Auerbach

Haftungsausschluß

Der Inhalt dieser Dokumentation wurde sorgfältig hinsichtlich seiner Richtigkeit und der Übereinstimmung mit der Hardware geprüft. Abweichungen, gerade im Sinne der Produktverbesserung können jedoch nicht gänzlich ausgeschlossen werden, sodaß wir keine Gewähr auf vollständige Übereinstimmung übernehmen können. Diese Dokumentation unterliegt der regelmäßigen Überprüfung; anfallende Korrekturen werden in Folgeversionen enthalten sein. Bitte richten Sie Ihre Verbesserungsvorschläge an die u.g. Adresse. Technische Verbesserungen behalten wir uns vor.

Warenzeichen

- Raspberry Pi und Raspberry Pi Logo sind eingetragene Warenzeichen der Raspberry Pi Foundation.
- Debian ist ein eingetragenes Warenzeichen der Software in the Public Interest, Inc. (SPI).
- Linux ist eingetragenes Warenzeichen von Linus Torvalds.
- Mean Well und deren Logo sind eingetragene Warenzeichen der Mean Well Enterprises Co., Ltd..

Die Rechte der vorgenannten Firmen sowie die Waren und Warennamen liegen bei diesen Firmen.

Kontakt

FTL – Auerbach
Inhaber Dag Auerbach
St.-Vitus-Str. 41
82205 Gilching

Tel.: +49 (0)176 45964329
eMail: FTL-Auerbach@gmx.de

Inhalt

1	Allgemeines.....	5
2	Sicherheitshinweise.....	6
3	Hubo-Maxi Hardware Rev. 1.00.....	7
3.1	Technische Daten.....	7
3.2	Stückliste.....	8
3.3	Schaltplan.....	9
3.4	Arbeitsreihenfolge während der Bestückung.....	10
3.4.1	Das Board.....	10
3.4.2	Bestückung der Keramikkondensatoren und des Widerstandsnetzwerkes.....	10
3.4.3	IC Fassungen und MOSFET Sockel.....	11
3.4.4	Wannenstecker sowie Stift- und Buchsenleisten.....	11
3.4.5	Anschlußklemmen und Widerstände.....	12
3.4.6	Widerstände, Referenzspannungsquelle und Relaissteckoption.....	12
3.4.7	Elkos, Relais und gesockelte Halbleiter.....	13
3.4.8	Die fertig gelötete Unterseite und das konfektionierte Kabel.....	13
4	Hubo-Digital Hardware Rev. 1.00.....	15
4.1	Technische Daten.....	15
4.2	Stückliste.....	16
4.3	Schaltplan.....	17
4.4	Arbeitsreihenfolge während der Bestückung.....	18
4.4.1	Das Board.....	18
4.4.2	Bestückung der Keramikkondensatoren, der liegenden Widerstände, des Widerstandsnetzwerkes und des MOSFET Sockels.....	18
4.4.3	IC Fassungen, Stift- und Buchsenleisten.....	19
4.4.4	Anschlußklemmen und 1wire-Widerstand.....	19
4.4.5	Elko, Relaissteckoption, Relais und gesockelte Halbleiter.....	20
4.4.6	Die fertig gelötete Unterseite und das konfektionierte Kabel.....	20
5	Hubo-Maxi Hardware Rev. 1.11.....	22
5.1	Technische Daten.....	22
5.2	Stückliste.....	22
5.3	Schaltplan.....	23
5.4	Arbeitsreihenfolge während der Bestückung.....	24
6	Hubo-Maxi Hardware Rev. 2.01.....	25
6.1	Technische Daten.....	25
6.2	Stückliste.....	26
6.3	Schaltplan.....	27
6.4	Arbeitsreihenfolge während der Bestückung.....	28
7	Hubo-Maxi Hardware Rev. 3.00.....	29
7.1	Technische Daten.....	29
7.2	Stückliste.....	32
7.3	Schaltplan.....	33
7.4	Arbeitsreihenfolge während der Bestückung.....	34
8	Hardwarerevisionen.....	35
9	Inbetriebnahme der Hardware und Programmierung.....	36
9.1	Stromversorgung von Hubo und Raspberry Pi.....	36
9.2	Kaskadierung mehrerer Hubos.....	37

9.3	Referenzspannungserzeugung	38
9.4	Anschluß digitaler Sensoren	38
9.5	Analoge Spannungsmessung	39
9.6	Temperaturmessung mittels MCP9700/MCP9701	40
9.7	Anschluß von 1wire Modulen	40
9.7.1	Normalbetrieb an 3 Leitungen.....	41
9.7.2	Parasitärer Betrieb mit 2 Leitungen	42
9.7.3	Temperaturmessung mittels DS18x20	42
9.8	Die Programmierung mittels der Hubo C++-Library	43
9.8.1	Lesen und Schreiben der digitalen Kanäle	43
9.8.2	Lesen der analogen Kanäle	44
9.8.3	Inbetriebnahme mittels der Hubo C++-Library Demoprogramme	45
9.8.4	Generelle Anmerkungen zur Verwendung der Hubo C++-Library	46
9.9	Die Programmierung in Python und anderen Sprachen	47
9.9.1	Lesen und Schreiben der digitalen Kanäle in Python	47
9.9.2	Lesen der analogen Kanäle in Python	48
10	Empfehlungen	49
10.1	Hardwareergänzungen der Hubo-Produktreihen der Hardwarerevisionen 1 und 249	49
10.1.1	Hutschienengehäuse	49
10.1.2	Hutschienennetzteil	50
10.1.3	DS3231 Real Time Clock und 433MHz Sender	50
10.1.4	Externes Lastrelais zum Schalten größerer Lasten	51
10.1.5	Alle Optionen im Überblick.....	51
10.2	Hardwareergänzungen zur Hubo-Produktreihe der Hardwarerevision 3.....	52
10.2.1	Hutschienengehäuse	52
10.2.2	Hutschienennetzteil	52
10.2.3	DS3231 Real Time Clock und 433MHz Sender	53
10.3	1wire Schnittstelle.....	54
10.4	Sensoren	55

1 Allgemeines

Herzlichen Glückwunsch zum Erwerb des Hubo Bausatzes.

Die Bausätze der Hubo Produktreihe stellen eine universelle Erweiterung zu diversen Mikrocontrollern, u.a. auch zur GPIO Schnittstelle des Raspberry Pi dar. Sie sind kompatibel zu den Modellen Zero, A+, B, B+, 2 B, 3 B, 3 B+ und 4 B.

Ziel der Bausätze ist es, eine einfache Hardwarebasis zu schaffen, um den theoretischen und praktischen Umgang mit Mikrocontrollern und deren Peripherie für schul- und Lehrzwecke zu vertiefen.

Insofern wurde bei der Entwicklung besonderes Augenmerk darauf gelegt, dass die Lötaufgaben sehr einfach durchzuführen sind. Ebenso wurden die Bauteile so dimensioniert, dass nur wenige unterschiedliche Komponenten Einsatz finden. Dennoch konnte das Layout so gestaltet werden, dass die Messgenauigkeit des 12bit AD Wandlers (für den Fall eines Hubos mit AD Wandler – z.B. Hubo-Maxi) auch bei nicht idealen EMV Bedingungen voll ausgenutzt werden kann. Die auf dem Board befindliche Referenzspannungsquelle erlaubt Messauflösungen von bis zu 0,61mV. Sie ist ideal angepasst zur direkten Verwendung z.B. der Temperatursensoren MCP9700 und MCP9701.

Alle empfindlichen Halbleiterkomponenten wurden gesockelt. Im Zuge eines Defektes lassen sich die Komponenten somit ohne weitere Lötarbeit schnell austauschen.

Hubo bietet ein offenes Konzept. So gibt es nicht nur unterschiedliche Ausführungen (Hubo-Maxi, Hubo-Digital, Hubo-Opto) sondern auch einen nach außen geführten I2C Bus mittels dessen sich je nach Modul bis zu 7 Slavemodule nachrüsten lassen. In der maximalen Ausbaustufe ergeben sich somit 8 analoge und 64 digitale Eingänge, sowie 64 digitale Ausgänge. Der 1wire Bus erlaubt den komfortable Anschluss vieler 1wire Produkte. Die Platinen gestatten die Nachrüstung verschiedenster Zusatzmodule wie batteriegepufferte Echtzeituhren (DS3231), Lastrelaiskarten oder (433MHz) FM-Transmitter.

Auch die Gehäusefrage wird von der Hubo Produktpalette gelöst. So kann Hubo in ein handelsübliches Hutschienengehäuse (siehe Hardwareergänzungen) montiert werden. Die Modelle Zero, A+, B+, 2 B, 3 B, 3 B+ und 4 B können zudem im Gehäusedeckel untergebracht werden.

Aufgrund der Verwendung von Standardhardware lässt sich Hubo mittels gängiger Bibliotheken (z.B. Python) steuern. Sollte die Performanz der Scriptsprachen den Anforderungen nicht genügen, so kann auf die Hubo C++-Library zurückgegriffen werden.

Bitte lesen Sie die nachfolgende Anleitung sorgfältig durch und beachten Sie insbesondere die Sicherheitshinweise und Anmerkungen zur Arbeitsreihenfolge bei der Bestückung der Platine.

Wir wünschen Ihnen viel Freude mit Ihrem Hubo Bausatz.

2 Sicherheitshinweise

Lesen Sie den folgenden Text vor dem Zusammenbau des Hubos aufmerksam durch. Unterrichten Sie alle Personen, welche Hubo verwenden über den Inhalt dieses Dokumentes. Dies gilt insbesondere beim Einsatz für Lehr und Demonstrationszwecke.

Die Nichtbeachtung der Hinweise der technischen Dokumentation, insbesondere auch dieser Sicherheitshinweise führt zum Erlöschen der Gewährleistungsansprüche für diesbezüglich begründete Schäden. Die Firma FTL – Auerbach, Inhaber Dag Auerbach übernimmt in diesen Fällen keine Haftung!

Hubo darf nicht im produzierenden Gewerbe eingesetzt werden. Die Verwendung in einem Umfeld, in dem Sach- und Personenschäden eintreten können, geschieht auf eigene Verantwortung des Betreibers.

Aus Sicherheitsgründen, insbesondere jedoch aus Gründen des Personenschutzes, ist ein Betrieb unter höheren Spannungen als 5-24V (bausatzabhängig, siehe Platinenaufdruck) nicht zulässig. Es wird ausdrücklich darauf hingewiesen, dass Arbeiten an Spannungen über 40V lebensgefährlich sein können. Verwenden Sie zum Betrieb Hubos nur Netzteile, welche die gesetzlichen Regelungen erfüllen (z.B. CE- und VDE-Zeichen).

Der Einsatz Hubos, sowie dessen Montage sind nur in trockenen und sauberen Räumen zulässig. Betrieb und Montage im Freien bzw. in Feuchträumen, staubbelasteten oder explosionsgefährdeten Umgebungen werden ausdrücklich ausgeschlossen. Die zulässige Betriebstemperatur liegt bei 10°C bis 40°C.

Arbeiten am Hubo dürfen nur im stromlosen Zustand erfolgen. Nach dem Betrieb des Hubo ist dessen Stromzufuhr zu unterbrechen.

Zur Reparatur dürfen nur Originalersatzteile verwendet werden.

Der Einsatz in Ausbildungseinrichtungen und Schulen ist von geschultem Fachpersonal zu begleiten.

Bitte beachten Sie, dass der Bausatz leicht zu verschluckende und spitze Kleinteile enthält, welche nicht in Kinderhand gehören. Ebenso ist beim Umgang mit dem LötKolben Vorsicht geboten, da hierdurch schwerwiegende Verbrennungen entstehen können.

Der Bausatz enthält empfindliche elektronische Bauteile, welche vor elektrostatischen Entladungen zu schützen sind. Dazu zählen insbesondere die IC's und der MOSFET.

3 Hubo-Maxi Hardware Rev. 1.00

Hubo-Maxi stellt die voll ausgebaute Variante des Hubo-Boards dar.

3.1 Technische Daten

Die Hubo – Maxi Masterplatine verfügt über die folgenden Eckdaten:

- Platinengröße (LxB): 100mm x 85mm
- Gewicht: 90g
- Versorgungsspannung: 5V
- 8 x analoge Eingänge mit einer Referenzspannung von $U_{Ref}=2,5V$ und einer Auflösung von 4096 Digit (entspricht 0,61mV)
- 8 x digitale Eingänge (mit Pullup-Widerstandsnetzwerk bestückt)
- 4 x digitale Relaisausgänge in Bestückungsvarianten von 3/5A 250VAC
- 3 (4) x Open Collector Ausgänge 500mA/50V (bei Verzicht auf den 1wire Ausgang stehen 4 Open Collector Ausgänge zur Verfügung)
- 1 x 1wire Ausgang mit 4,7kOhm Pullup-Widerstand
- 1 x Steckoption für Real Time Clock Modul DS3231 (die Module selbst können kostengünstig im Internet bezogen werden und sind nicht Bestandteil des Lieferumfangs)
- 1 x Steckoption für 8 Kanal Lastrelaiskarte 10A, 250VAC (die Relaiskarte kann kostengünstig im Internet bezogen werden und sind nicht Bestandteil des Lieferumfangs)
- 1 x I2C Ausgang (SDA, SCL, V_{DD} , GND) zur Kaskadierung weiterer Module

Der Steuerbaustein der digitalen Ein- und Ausgänge erlaubt eine Adresselektion, so dass sich bis zu 3 weitere Module kaskadieren lassen (auf dem Master bedarf es keiner Bestückung der Adresselektoren).

Der Formfaktor wurde so gewählt, dass er den Einbau in ein handelsübliches Hutschienengehäuse erlaubt (siehe Hardwareergänzungen).

3.2 Stückliste

Stückliste des Hubo-Maxi Hardware Rev. 1.00 Mastermoduls:

- 1 x Hubo Master-Platine (100x85; 35µ Cu; doppelseitig durchkontaktiert, Lötstopplack, verzinkt)
- 35 x Anschlussklemmen (2- und 3-polig)
- 8 x Widerstand 470Ω (R1..R8)
- 1 x Widerstand 32-330Ω (R9)
- 3 x Widerstand 4,7kΩ (R10, R11, R100)
- 1 x Widerstandsnetzwerk 470Ω-4,7kΩ-10kΩ (RN1; je nach Einsatzzweck bestückt)
- 11 x Kondensator 100nF (C1..C8, C10, C12, C13)
- 1 x Kondensator 100µF-220µF (C9)
- 1 x Kondensator 22µF (C11)
- 1 x MOSFET 2N7000 (T100)
- 1 x TL431 (Spannungsreferenz)
- 1 x MCP3208 (12 bit AD Wandler)
- 1 x MCP23017 (IO Expander)
- 1 x ULN2803 (Treiber IC)
- 1 x Stiftleiste 3 polig (JP100)
- 1 x Jumper
- 1 x Buchsenleiste 3 polig (Sockel für 2N7000; T100)
- 1 x Buchsenleiste 6 polig (RTC-MEM-Modul)
- 1 x Stiftleiste 10 polig (JP3 für externe Lastrelaisplatine)
- 1 x Wannenstecker 2x13 polig (JP4)
- 1 x IC Sockel 28 polig (IC1)
- 1 x IC Sockel 18 polig (IC2)
- 1 x IC Sockel 16 polig (IC3)
- 4 x Relais 3-5A 250VAC Schließer (verschiedene Ausführungen mit mind. 3A Schaltstrom)
- 4 x Abstandhalter für Hutschienegehäuse
- 4 x Mutter M3
- 4 x Schraube M3x12
- 2 x Pfostenbuchse 26 polig (für Hubo und Raspberry Pi B)
- 1 x Pfostenbuchse 40 polig (im Falle der Verwendung eines Raspberry Pi Zero, A+, B+, 2 B, 3 B, 3 B+)
- 1 x Flachbandkabel

Achtung, die beiden zur Adressselektion des IO Expanders vorgesehenen Jumper JP1 und JP2 werden auf dem Mastermodul nicht bestückt! Die Jumper werden nur für kaskadierte Slavemodule notwendig. Das Mastermodul läuft somit immer auf der I2C Adresse 0x20.

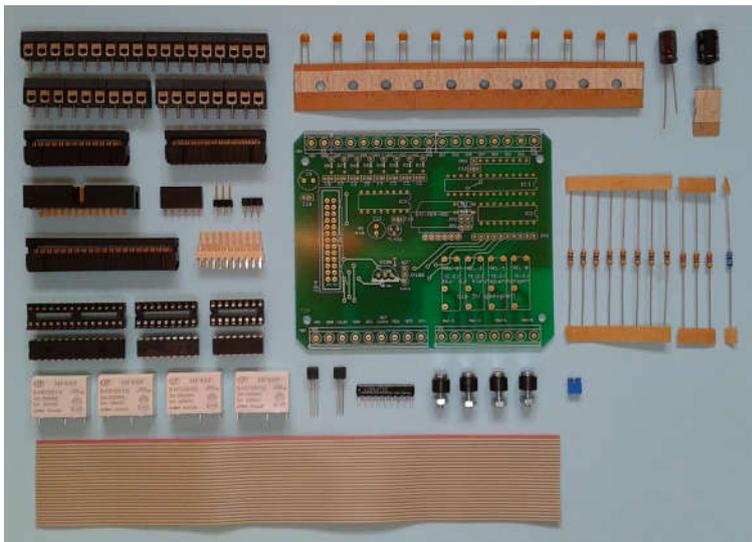


Bild 1: Bauteile des Hubo-Maxi Hardware Rev. 1.00 Mastermoduls

3.3 Schaltplan

Den Schaltplan des Hubo-Gesamtmoduls (Vollausbau inkl. Master- und Slavemodul) zeigt das folgende Bild. Sehen sie auch dazu die Datei *Hubo_Schaltplan_1.00.03.pdf*.

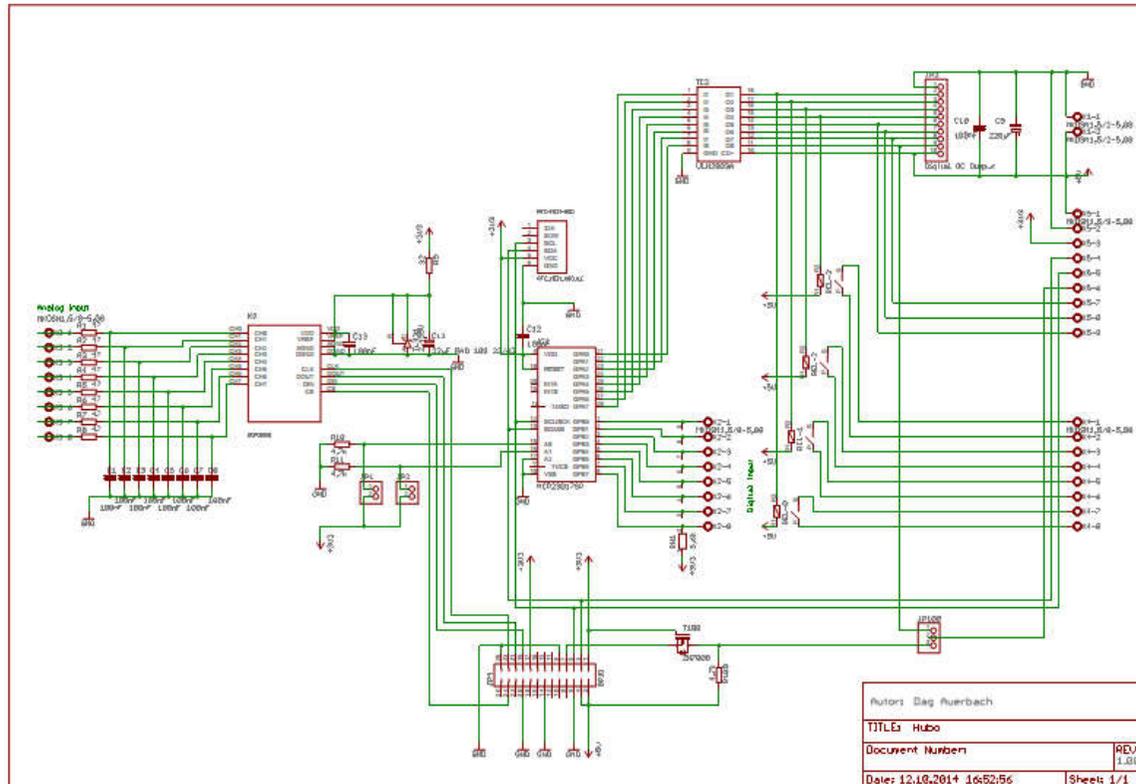


Bild 2: Schaltplan des Hubo-Gesamtmoduls (Hardware Rev. 1.00)

3.4 Arbeitsreihenfolge während der Bestückung

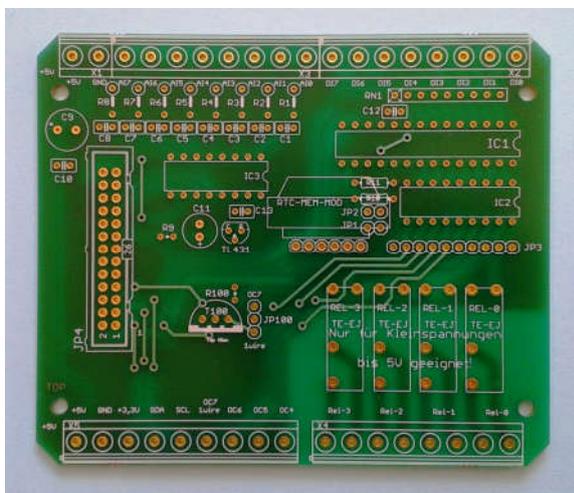
Im Sinne eines einfacheren und schnellen Aufbaus empfiehlt es sich, die Bestückung der Platine in einer gewissen Reihenfolge vorzunehmen. Grundsätzlich ist es sinnvoll, zuerst mit den Bauteilen der geringsten Höhe anzufangen. Damit ist gewährleistet, dass das Bauteil beim Löten weniger verrutschen kann.

Zur Abwinklung der Widerstände sollte eine kleine Rundzange verwendet werden.

3.4.1 Das Board

Die Bilder 3 und 4 zeigen die Ober- und Unterseite der Platine. Die Oberseite enthält die Bestückungsbeschriftung zur leichteren Platzierung der Bauelemente.

Anmerkung: Bei der hier abgebildeten Platine handelt es sich um einen Prototyp. In der Regel werden die Platinen verzinkt geliefert.



3.4.3 IC Fassungen und MOSFET Sockel

Beim Einlöten der IC Fassungen beginnt man mit der größten.

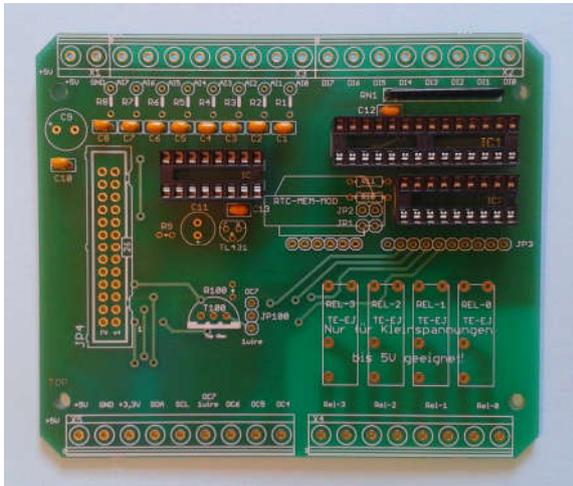


Bild 7: 3 IC-Fassungen ...

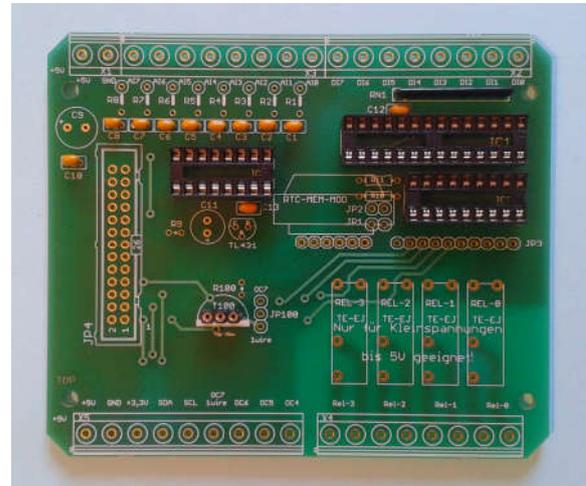


Bild 8: ... und Sockel für MOSFET

3.4.4 Wannenstecker sowie Stift- und Buchsenleisten

Bei den Stift- und Buchsenleisten sollte zunächst nur ein Pin eingelötet werden. Somit kann man deren Ausrichtung kontrollieren und ggf. noch korrigieren.

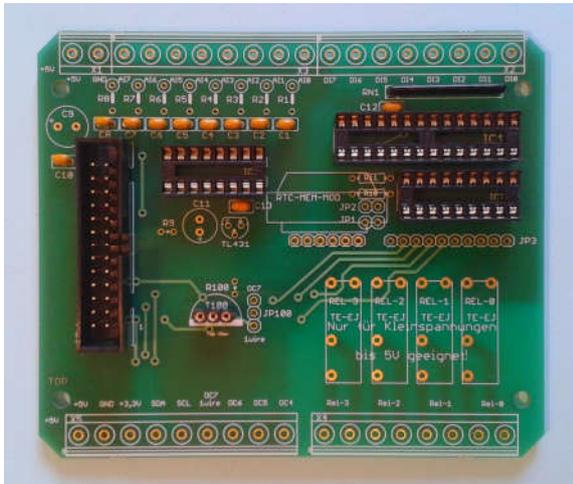


Bild 9: Wannenstecker ...

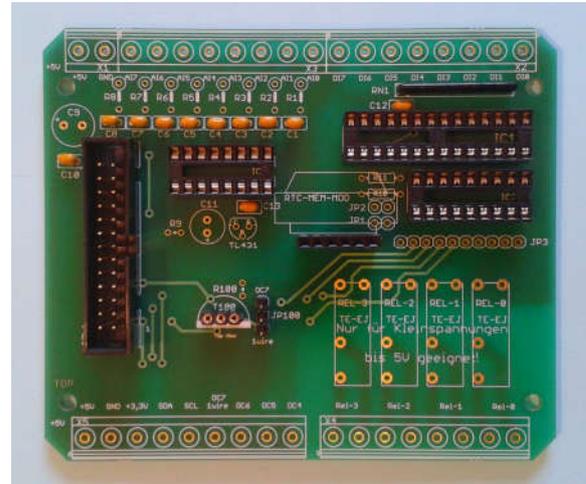


Bild 10: ... und Stift- und Buchsenleisten

3.4.5 Anschlußklemmen und Widerstände

Auch beim Einlöten der Anschlussklemmen sollte man zunächst einen einzelnen mittleren Pin fixieren, um ggf. noch korrigieren zu können. Es empfiehlt sich, die Platine auf den Kopf zu legen und einen leichten Druck auf sie auszuüben, damit die Klemmen ohne Abstand auf der Platine aufliegen.

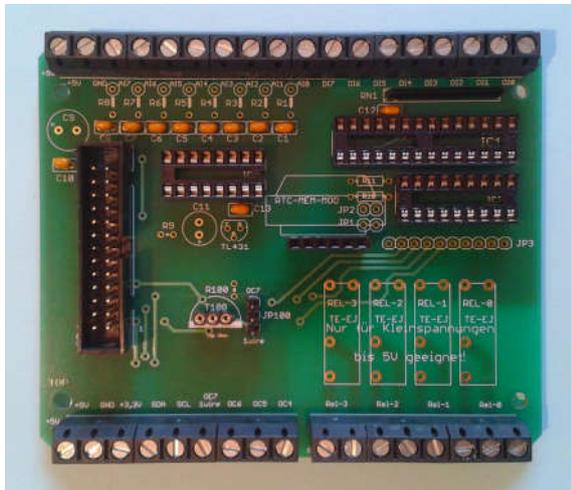


Bild 11: Anschlussklemmen ...

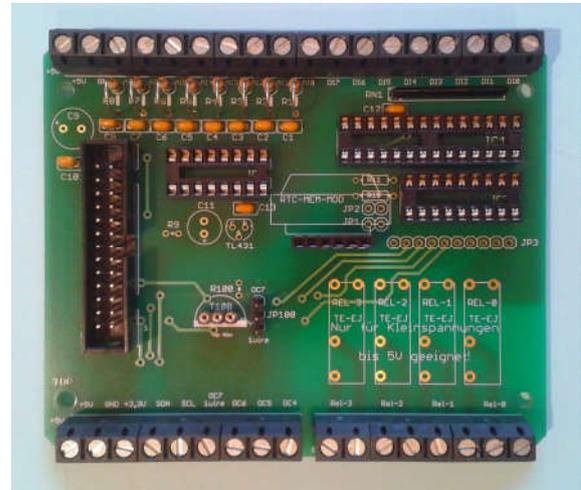


Bild 12: ... und höhenausgerichtete Widerstände

Die Anschlussklemmen sind die bis jetzt höchsten Bauteile und stellen damit eine gleichmäßige Höhenausrichtung dar, wenn die Platine auf den Kopf gelegt wird. Das erlaubt eine einfache Höhenausrichtung der nun folgenden Bauteile (z.B. der Widerstände). Verwechseln sie die Widerstände nicht!

3.4.6 Widerstände, Referenzspannungsquelle und Relaissteckoption

Die Höhenausrichtung der beiden verbleibenden stehenden Widerstände, erfolgt analog zu denen des Analogteiles. Achten Sie darauf, dass die beiden anderen Widerstände zur Adresselektion liegend montiert werden. Achten Sie auch hier auf die Widerstandswerte.

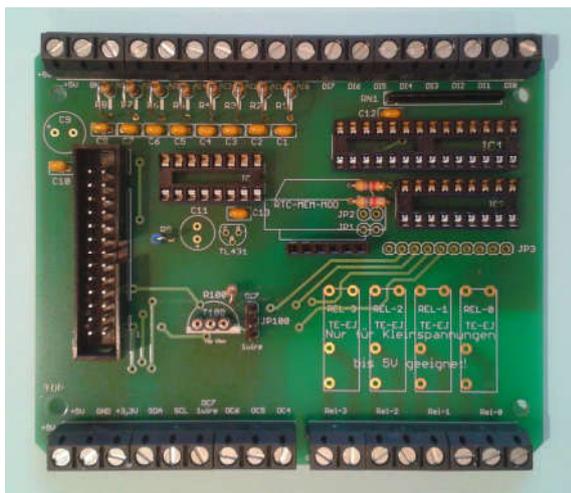


Bild 13: Weitere Widerstände ...

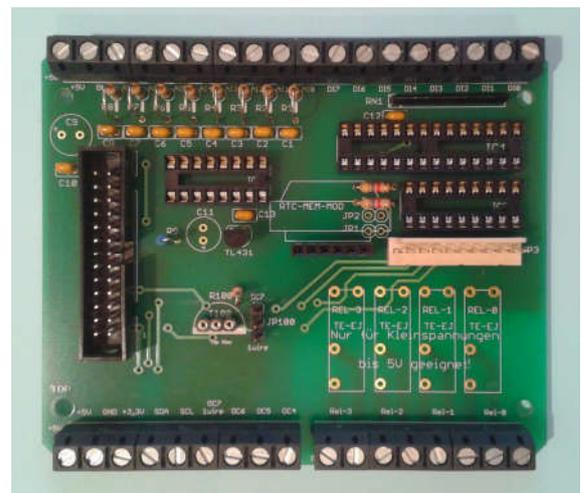


Bild 14: ... Referenzspannungsquelle und Relaissteckoption

Achtung, verwechseln Sie die Referenzspannungsquelle TL431 nicht mit dem MOSFET oder anderen Bauteilen im TO92-Gehäuse!

3.4.7 Elkos, Relais und gesockelte Halbleiter

Da sowohl die Elkos, als auch die Relais die Höhe der Anschlußklemmen überschreiten, lässt sich das Board danach nicht mehr kippfrei auf den Kopf legen. Daher werden diese Komponenten als letzte eingelötet.

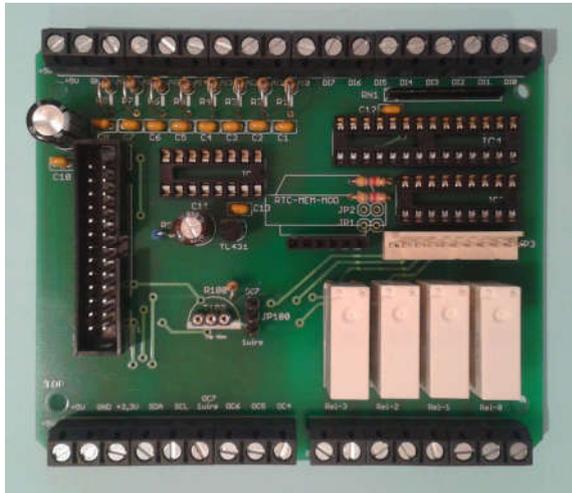


Bild 15: Elkos und Relais...



Bild 16: ... und gesockelte Halbleiter

Achten Sie beim Einstecken der Halbleiter auf ESD Schutz! Vergessen Sie nicht den Jumper JP100 entweder auf OC oder 1wire zu stellen, je nachdem ob Sie den 1wire Bus benötigen oder nicht.

3.4.8 Die fertig gelötete Unterseite und das konfektionierte Kabel

Bild 17 zeigt die vollständig bestückte Platine von der Lötseite. Beachten Sie, dass die Jumper JP1 und JP2 beim Mastermodul nicht bestückt werden.

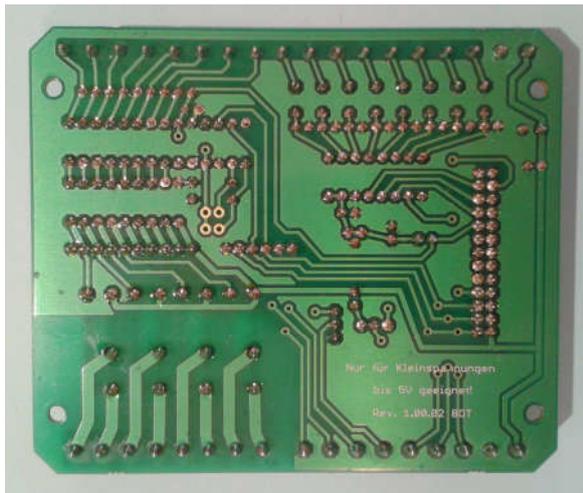


Bild 17: Unterseite...



Bild 18: ... und konfektionierte Kabel

Die Pfostenbuchsen werden am besten in einem Schraubstock auf die Kabel geklemmt. Das gewährleistet einen gleichmäßigen Druck und ein paralleles Schließen der Buchsen. Achten Sie unbedingt auf die richtige Lage der Ader 1 (rot) und die Nasen der Pfostenbuchsen. Im Falle des Einsatzes eines Raspberry Zero, A+, B+, 2 B, 3 B, 3 B+ bzw. 4 B im Deckel des Hutschienengehäuses bedarf es einer Z-artigen Montage der Buchsen (siehe Bild 17), damit das Kabel zwischen den beiden Platinen verlegt werden kann. Die ideale Gesamtlänge des Kabels beträgt dabei 120mm.

4 Hubo-Digital Hardware Rev. 1.00

Hubo-Digital enthält die Erweiterungen der digitalen Schnittstelle des Hubo-Boards. Selbstverständlich kann der analoge Eingangsteil später nachgerüstet werden.

4.1 Technische Daten

Die Hubo – Maxi Masterplatine verfügt über die folgenden Eckdaten:

- Platinengröße (LxB): 100mm x 85mm
- Gewicht: 80g
- Versorgungsspannung: 5V
- 8 x digitale Eingänge (mit Pullup-Widerstandsnetzwerk bestückt)
- 4 x digitale Relaisausgänge in Bestückungsvarianten von 3/5A 250VAC
- 3 (4) x Open Collector Ausgänge 500mA/50V (bei Verzicht auf den 1wire Ausgang stehen 4 Open Collector Ausgänge zur Verfügung)
- 1 x 1wire Ausgang mit 4,7kOhm Pullup-Widerstand
- 1 x Steckoption für Real Time Clock Modul DS3231 (die Module selbst können kostengünstig im Internet bezogen werden und sind nicht Bestandteil des Lieferumfangs)
- 1 x Steckoption für 8 Kanal Lastrelaiskarte 10A, 250VAC (die Relaiskarte kann kostengünstig im Internet bezogen werden und sind nicht Bestandteil des Lieferumfangs)
- 1 x I2C Ausgang (SDA, SCL, V_{DD}, GND) zur Kaskadierung weiterer Module

Der Steuerbaustein der digitalen Ein- und Ausgänge erlaubt eine Adresselektion, so dass sich bis zu 3 weitere Module kaskadieren lassen (auf dem Master bedarf es keiner Bestückung der Adresselektoren).

Der Formfaktor wurde so gewählt, dass er den Einbau in ein handelsübliches Hutschienengehäuse erlaubt (siehe Hardwareergänzungen).

4.3 Schaltplan

Für eine Übersicht zum Hubo Gesamtschaltplan schauen Sie bitte beim Hubo-Maxi nach. Sehen sie auch dazu die Datei *Hubo_Schaltplan_1.00.03.pdf*.

4.4 Arbeitsreihenfolge während der Bestückung

Im Sinne eines einfacheren und schnellen Aufbaus empfiehlt es sich, die Bestückung der Platine in einer gewissen Reihenfolge vorzunehmen. Grundsätzlich ist es sinnvoll, zuerst mit den Bauteilen der geringsten Höhe anzufangen. Damit ist gewährleistet, dass das Bauteil beim Löten weniger verrutschen kann.

4.4.1 Das Board

Die Bilder 20 und 21 zeigen die Ober- und Unterseite der Platine. Die Oberseite enthält die Bestückungsbeschriftung zur leichteren Platzierung der Bauelemente. Hubo Platinen werden verzinkt und mit Lötstopplack geliefert.

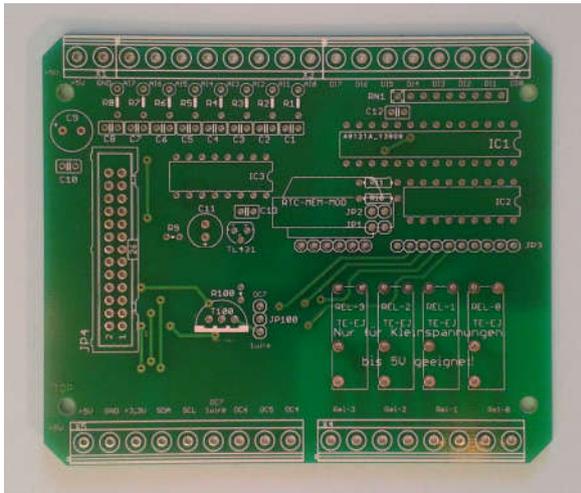


Bild 20: Board Oberseite

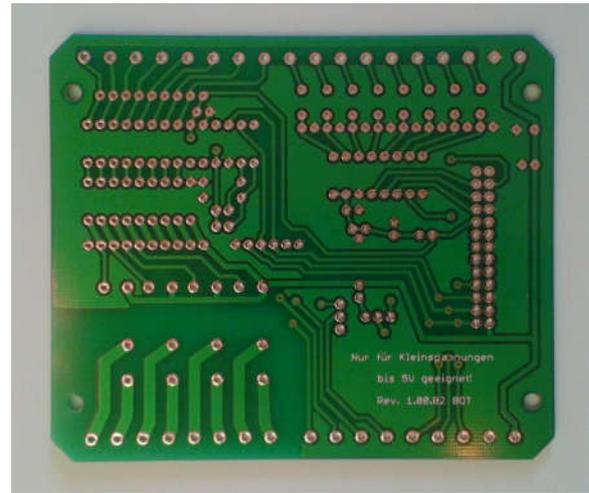


Bild 21: Board Unterseite

4.4.2 Bestückung der Keramikkondensatoren, der liegenden Widerstände, des Widerstandsnetzwerkes und des MOSFET Sockels

Zunächst werden die Keramikkondensatoren und die liegenden Widerstände bestückt, danach folgen das Widerstandsnetzwerk und der MOSFET Sockel.

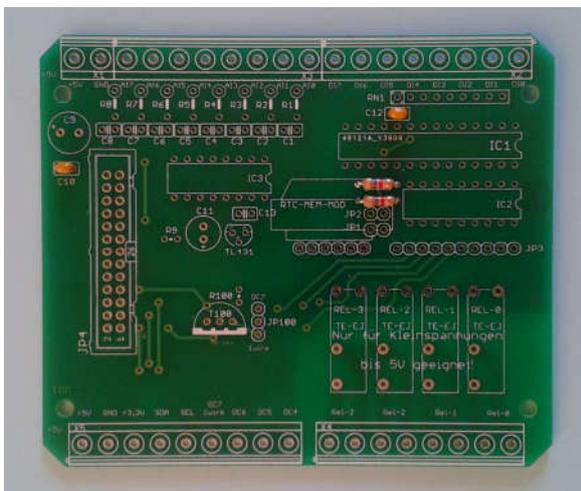


Bild 22: Kondensatoren, Widerstände...

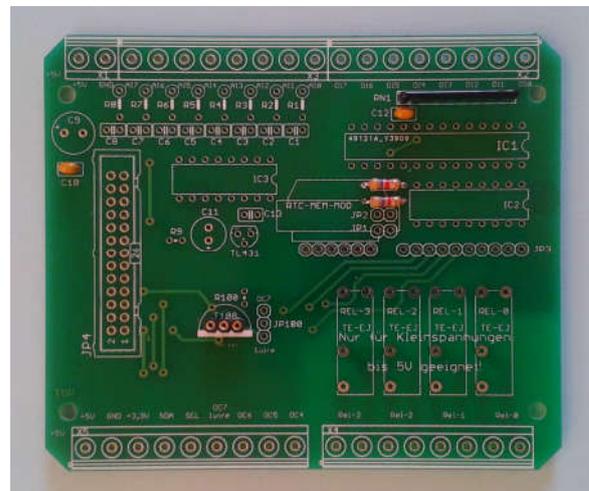


Bild 23: ... Widerstandsnetzwerk und MOSFET Sockel

4.4.3 IC Fassungen, Stift- und Buchsenleisten

Beim Einlöten der IC Fassungen beginnt man mit der größeren zuerst. Bei den nun folgenden Stift- und Buchsenleisten sollte zunächst nur ein Pin eingelötet werden. Somit kann man deren Ausrichtung kontrollieren und ggf. noch korrigieren.

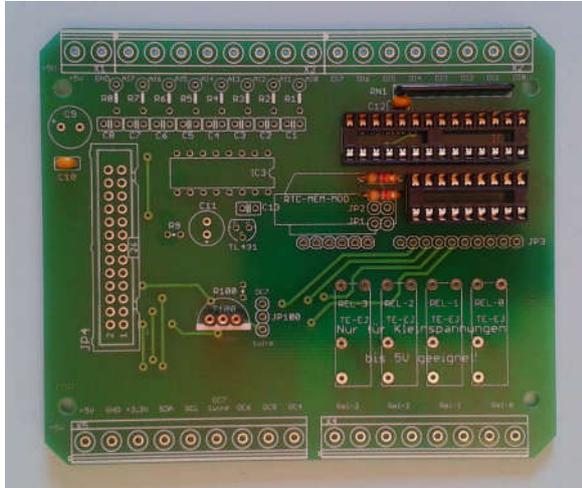


Bild 24: IC-Fassungen, ...

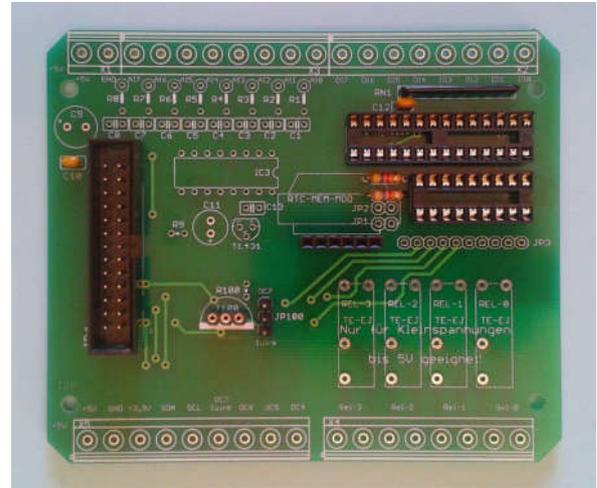


Bild 25: ... und Stift- und Buchsenleisten

4.4.4 Anschlußklemmen und 1wire-Widerstand

Auch beim Einlöten der Anschlussklemmen sollte man zunächst einen einzelnen mittleren Pin fixieren, um ggf. noch korrigieren zu können. Es empfiehlt sich, die Platine auf den Kopf zu legen und einen leichten Druck auf sie auszuüben, damit die Klemmen ohne Abstand auf der Platine aufliegen.

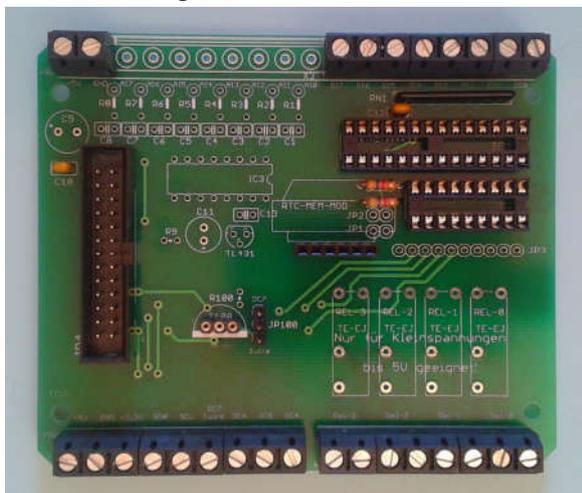


Bild 26: Anschlussklemmen ...

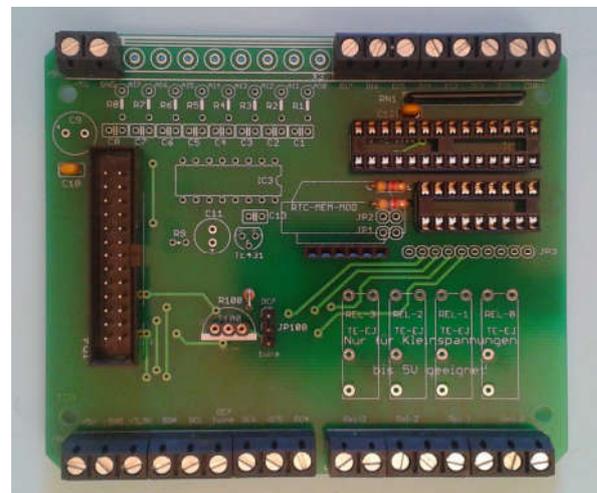


Bild 27: ... und höhenausgerichteter 1wire Widerstand

Die Anschlussklemmen sind die bis jetzt höchsten Bauteile und ermöglichen damit eine gleichmäßige Höhenausrichtung, wenn die Platine auf den Kopf gelegt wird. Das erlaubt ein einfaches Verlöten des stehenden Widerstands.

4.4.5 Elko, Relaissteckoption, Relais und gesockelte Halbleiter

Da sowohl der Elko, als auch die Relais die Höhe der Anschlußklemmen überschreiten, lässt sich das Board danach nicht mehr kippfrei auf den Kopf legen. Daher werden diese Komponenten als letzte eingelötet.

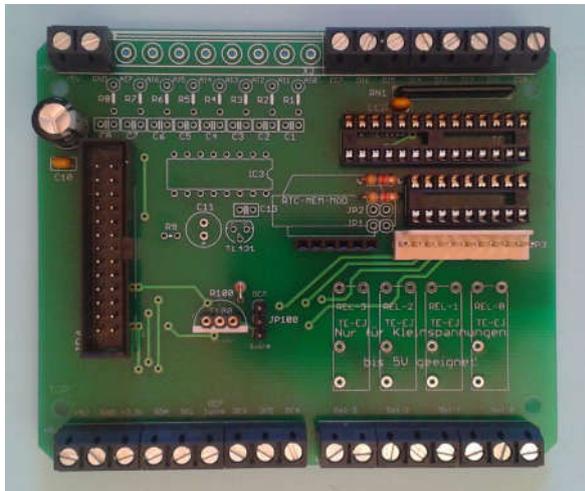


Bild 28: Elko, Relaissteckoption...

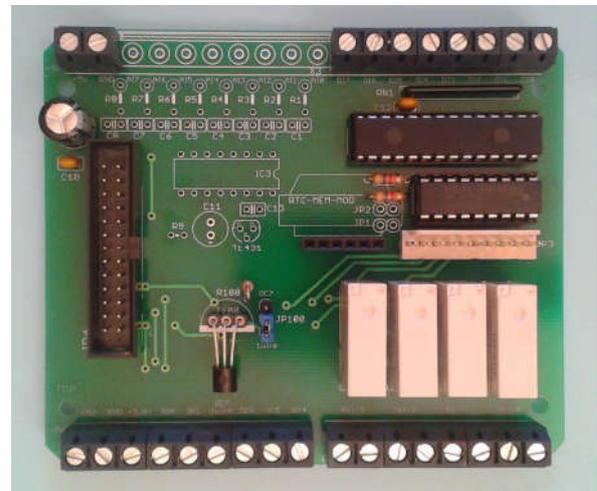


Bild 29: ... Relais und gesockelte Halbleiter

Achten Sie beim Einstecken der Halbleiter auf ESD Schutz! Vergessen Sie nicht den Jumper JP100 entweder auf OC oder 1wire zu stellen, je nachdem ob Sie den 1wire Bus benötigen oder nicht.

4.4.6 Die fertig gelötete Unterseite und das konfektionierte Kabel

Bild 30 zeigt die vollständig bestückte Platine von der Lötseite. Beachten Sie, dass die Jumper JP1 und JP2 beim Mastermodul nicht bestückt werden.

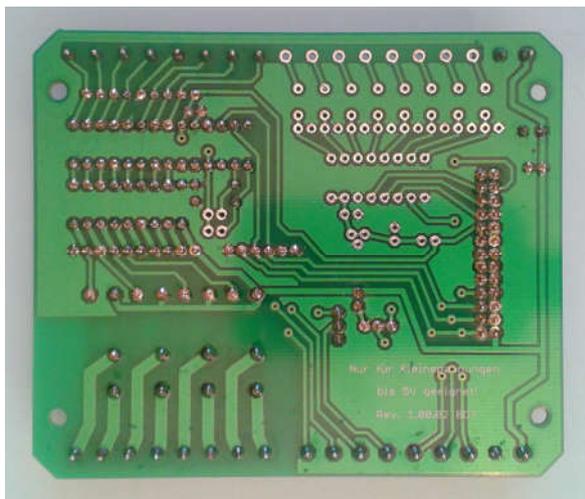


Bild 30: Unterseite...



Bild 31: ... und konfektionierte Kabel

Die Pfostenbuchsen werden am besten in einem Schraubstock auf die Kabel geklemmt. Das gewährleistet einen gleichmäßigen Druck und ein paralleles Schließen der Buchsen. Achten Sie unbedingt auf die richtige Lage der Ader 1 (rot) und die Nasen der Pfostenbuchsen. Im Falle des Einsatzes eines Raspberry Pi Zero, A+, B+, 2 B, 3 B, 3 B+ bzw. 4 B im Deckel des Hutschienengehäuses bedarf es einer Z-artigen Montage der Buchsen (siehe Bild 17), damit das Kabel zwischen den beiden Platinen verlegt werden kann. Die ideale Gesamtlänge des Kabels beträgt dabei 120mm.

5 Hubo-Maxi Hardware Rev. 1.11

Hubo-Maxi Hardware Rev. 1.11 wurde auf Basis der Hardware Rev. 1.00 entwickelt und um einige Funktionalitäten erweitert. Details hierzu entnehmen Sie bitte den technischen Daten.

5.1 Technische Daten

Die technischen Daten der Hardware Rev. 1.11 entsprechen denen der Hardware Rev. 1.00 mit den folgenden Erweiterungen:

- Polyfuse Sicherung zum Schutz vor Kurzschlüssen
- Überspannungsschutz für den 1wire Bus
- Lötoption für ein 433MHz Funkmodul z.B. zur Steuerung von Funksteckdosen
- Lötoption für Varistoren zum Schutz der Relaiskontakte (nicht im Lieferumfang inbegriffen)
- Die Kaskadierung der Module kann nun wahlweise nicht nur per Flachbandkabel sondern auch über die Schraubklemmen erfolgen. Der zusätzliche Gehäusedurchbruch für das Flachbandkabel entfällt damit.
- Vergrößerung der Schirmflächen des Bottom-Layers der Platine

5.2 Stückliste

Die Stückliste des Hubo-Maxi Mastermoduls Hardware Rev. 1.11 entspricht derjenigen der Hardware Rev. 1.00 mit den folgenden zusätzlichen Teilen:

2 x schnelle Schottky Diode (D101, D102)
1 x Zenerdiode (D2)
1 x Widerstand 1000Ohm (R101)
1 x Polyfuse Sicherung (F1)

Die Steckoption des MOSFET Transistors (T100) entfällt, da dieser fest verlötet wird.

5.3 Schaltplan

Den Schaltplan des Hubo-Gesamtmoduls (Vollausbau inkl. Master- und Slavemodul) zeigt das folgende Bild. Sehen sie auch dazu die Datei *Hubo_Schaltplan_1.11.00.pdf*.

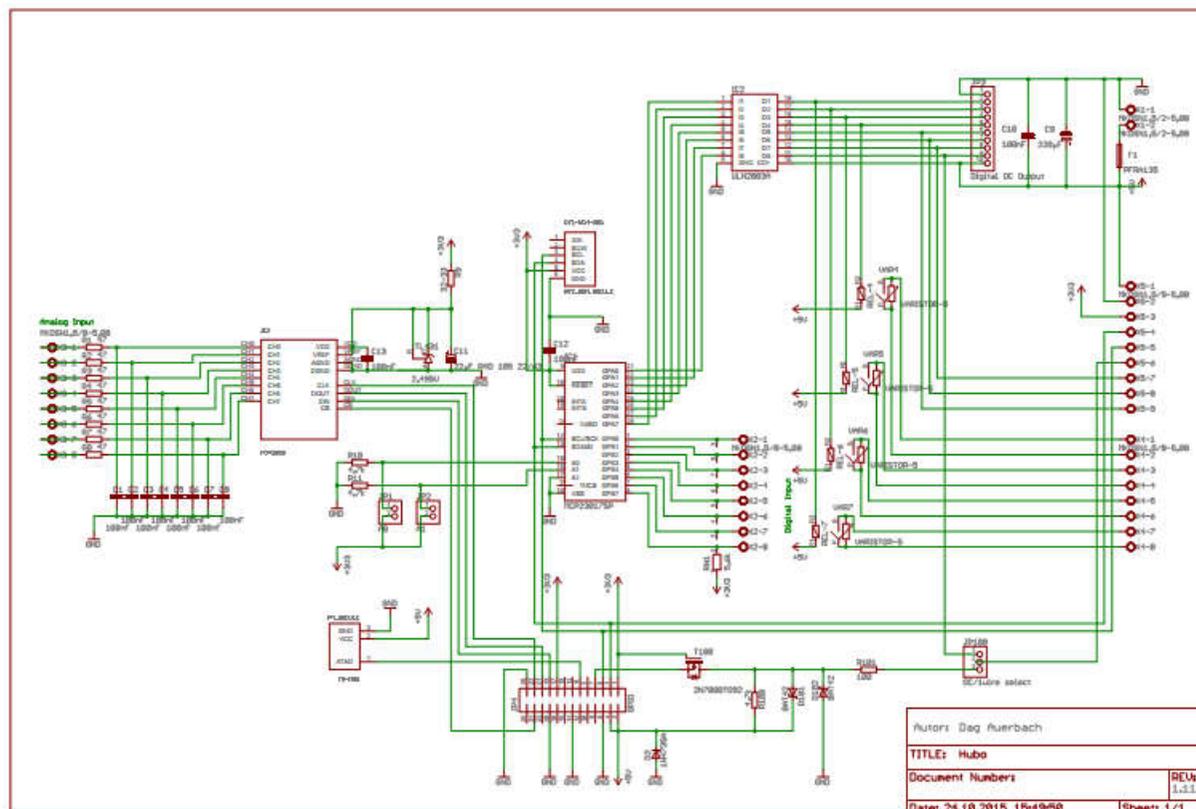


Bild 32: Schaltplan des Hubo-Gesamtmoduls (Hardware Rev. 1.11)

5.4 Arbeitsreihenfolge während der Bestückung

Bestücken Sie bei Hardware Rev. 1.11 zunächst die Dioden D101, D102 und D2. Danach folgen Sie bitte den Anweisungen wie unter der Hardware Rev. 1.00 beschrieben. Als letztes löten Sie die Polyfuse Sicherung F1 auf das Board. Bild 33 zeigt die vollständig bestückte Platine der Hardware Rev. 1.11.

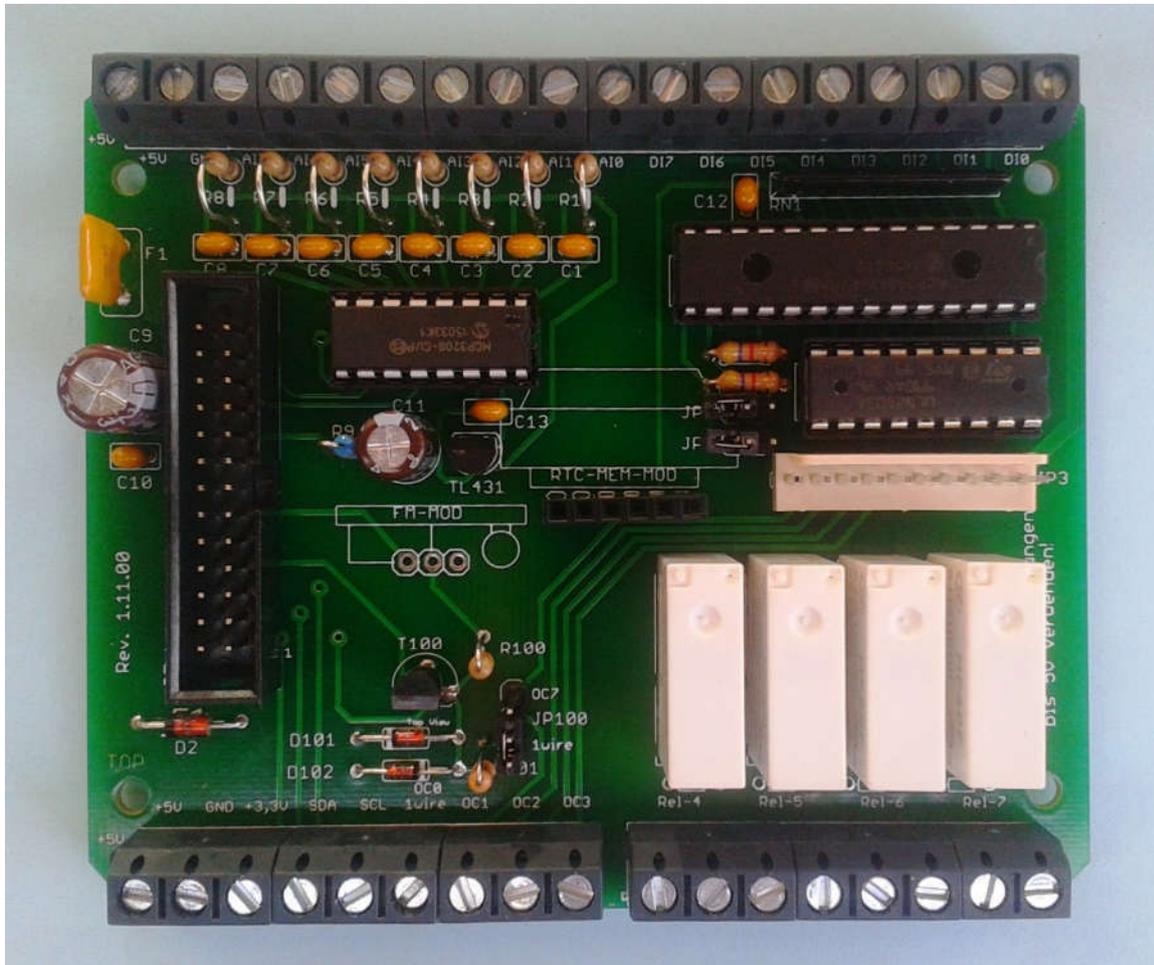


Bild 33: Hubo-Maxi Hardware Rev. 1.11 nach der Bestückung

Beachten Sie bitte, dass die Varistoren spannungsabhängig dimensioniert werden müssen und somit nicht zum Lieferumfang der Hubo Bausätze gehören!

6 Hubo-Maxi Hardware Rev. 2.01

Hubo-Maxi stellt die voll ausgebaute Variante des Hubo-Boards dar.

Um den Analogteil noch robuster von Störungen unterhalb des Hutschienenmoduls zu machen, wurden die Schaltkreise in SMD-Technik auf der Platinenoberseite positioniert. Damit gelingt eine geschlossene und schirmende Groundfläche auf deren. Gleichsam wurden analoger und digitaler Ground getrennt. AGND steht über eine separate Klemme zur Verfügung.

6.1 Technische Daten

Die Hubo – Maxi Masterplatine verfügt über die folgenden Eckdaten:

- Platinengröße (LxB): 100mm x 85mm
- Gewicht: 90g
- Versorgungsspannung: 5V
- 7 x analoge Eingänge mit einer Referenzspannung von $U_{Ref}=2,5V$ und einer Auflösung von 4096 Digit (entspricht 0,61mV)
- Separat verfügbarer Anschluß für den analogen Ground (AGND)
- 8 x digitale Eingänge (mit Pullup-Widerstandsnetzwerk bestückt)
- 4 x digitale Relaisausgänge in Bestückungsvarianten von 3/5A 250VAC
- 3 (4) x Open Collector Ausgänge 500mA/50V (bei Verzicht auf den 1wire Ausgang stehen 4 Open Collector Ausgänge zur Verfügung)
- 1 x 1wire Ausgang mit 4,7kOhm Pullup-Widerstand
- 1 x Steckoption für Real Time Clock Modul DS3231 (die Module selbst können kostengünstig im Internet bezogen werden und sind nicht Bestandteil des Lieferumfangs)
- 1 x Steckoption für 8 Kanal Lastrelaiskarte 10A, 250VAC (die Relaiskarte kann kostengünstig im Internet bezogen werden und sind nicht Bestandteil des Lieferumfangs)
- 1 x I2C Ausgang (SDA, SCL, V_{DD} , GND) zur Kaskadierung weiterer Module
- Polyfuse Sicherung zum Schutz vor Kurzschlüssen
- Überspannungsschutz für den 1wire Bus
- Lötoption für Varistoren zum Schutz der Relaiskontakte (nicht im Lieferumfang inbegriffen)
- Lötoption für Spannungsteiler zur Überwachung der Versorgungsspannung (die Widerstände R81 und R82 sind nicht im Lieferumfang inbegriffen)
- Die Kaskadierung der Module kann wahlweise per Flachbandkabel oder über die Schraubklemmen erfolgen. Der zusätzliche Gehäusedurchbruch für das Flachbandkabel entfällt damit.
- Vergrößerung der Schirmflächen des Bottom-Layers der Platine

Der Steuerbaustein der digitalen Ein- und Ausgänge erlaubt eine Adresselektion, so dass sich bis zu 3 weitere Module kaskadieren lassen (auf dem Master bedarf es jedoch keiner Bestückung der Adresselektoren).

Der Formfaktor wurde so gewählt, dass er den Einbau in ein handelsübliches Hutschienengehäuse erlaubt (siehe Hardwareergänzungen).

6.2 Stückliste

Stückliste des Hubo-Maxi Mastermoduls:

- 1 x Hubo Master-Platine (100x85; 35µ Cu; doppelseitig durchkontaktiert, Lötstopplack, verzinkt)
- 35 x Anschlussklemmen (2- und 3-polig)
- 7 x SMD-Widerstand 470Ω (R1..R7) vorbestückt
- 1 x Widerstand 32-33Ω (R9)
- 2 x SMD-Widerstand 4,7kΩ (R10, R11) vorbestückt
- 1 x Widerstand 4,7kΩ (R100)
- 1 x Widerstand 4,7kΩ (R81) nicht Bestandteil der Lieferumfangs
- 1 x Widerstand 6,5kΩ (R82) nicht Bestandteil der Lieferumfangs
- 1 x Widerstandsnetzwerk 470Ω-4,7kΩ-10kΩ (RN1; je nach Einsatzzweck bestückt)
- 9 x SMD-Kondensator 100nF (C1..C7, C12, C14) vorbestückt
- 2 x Kondensator 100nF (C10, C13)
- 1 x Kondensator 100µF-220µF (C9)
- 1 x Kondensator 22µF (C11)
- 1 x MOSFET 2N7000 (T100)
- 1 x TL431 (Spannungsreferenz)
- 1 x SMD-MCP3208 (12 bit AD Wandler) vorbestückt
- 1 x SMD-MCP23017 (IO Expander) vorbestückt
- 1 x ULN2803 (Treiber IC)
- 1 x Stiftleiste 3 polig (JP100)
- 2 x Stiftleiste 2 polig (JP1, JP2)
- 3 x Jumper
- 1 x Buchsenleiste 6 polig (RTC-MEM-Modul)
- 1 x Stiftleiste 10 polig (JP3 für externe Lastrelaisplatine)
- 1 x Wannenstecker 2x13 polig (JP4)
- 1 x IC Sockel 18 polig (IC2)
- 4 x Relais 3-5A 250VAC Schließer (verschiedene Ausführungen mit mind. 3A Schaltstrom)
- 4 x Abstandhalter für Hutschienengehäuse
- 4 x Mutter M3
- 4 x Schraube M3x12
- 2 x schnelle Schottky Diode (D101, D102)
- 1 x Zenerdiode (D2)
- 1 x Widerstand 100Ω (R101)
- 1 x Polyfuse Sicherung (F1)

Achtung, die beiden zur Adressselektion des IO Expanders vorgesehenen Jumper JP1 und JP2 werden auf dem Mastermodul nicht bestückt! Die Jumper werden nur für kaskadierte Slavemodule notwendig. Das Mastermodul läuft somit immer auf der I2C Adresse 0x20.

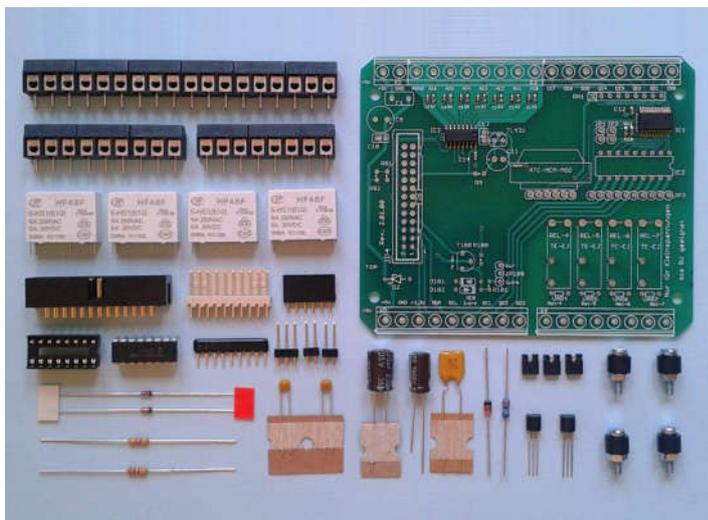


Bild 34: Bauteile des Hubo-Maxi Hardware Rev. 2.01 Mastermoduls

6.4 Arbeitsreihenfolge während der Bestückung

Bestücken Sie bei Hardware Rev. 2.01 zunächst die Dioden D101, D102 und D2. Danach folgen Sie bitte den Anweisungen wie unter der Hardware Rev. 1.00 beschrieben.

Überspringen Sie dabei die Hinweise zur Bestückung der bereits werkseitig vorbestückten SMD Bauteile. Als letztes löten Sie die Polyfuse Sicherung F1 auf das Board. Bild 36 zeigt die vollständig bestückte Platine der Hardware Rev. 2.01.

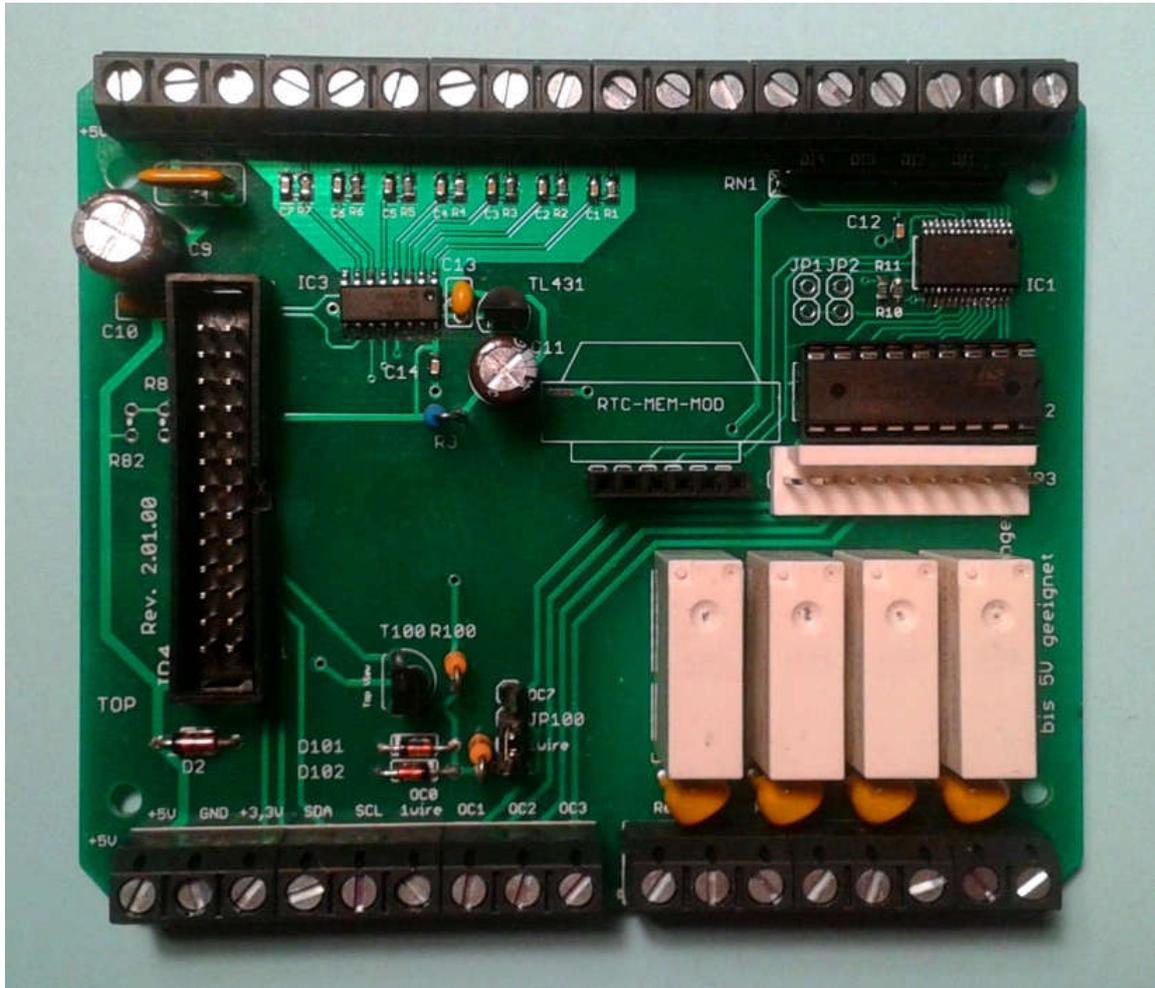


Bild 36: Hubo-Maxi Hardware Rev. 2.01 nach der Bestückung

Beachten Sie bitte, dass die Varistoren spannungsabhängig dimensioniert werden müssen und somit nicht zum Lieferumfang der Hubo Bausätze gehören!

7 Hubo-Maxi Hardware Rev. 3.00

Hubo-Maxi Hardware Rev. 3.00 ergänzt die Hubo Produktpalette hinsichtlich galvanisch getrennter 24V Digitalanschlüsse (sowohl Eingänge als auch Ausgänge). Zudem erlauben die Module Kaskadierungen auf insgesamt 8 Hubos (64 Eingänge und 64 Ausgänge). Hubo 3.00 ist hinsichtlich seiner gemeinsamen Funktionalitäten pin- und softwarekompatibel zu den Vorgängerversionen. Dies betrifft die Stromversorgung des Raspberry Pi, den I2C-Bus und den 1wire-Bus. Die Bestückungsoptionen für FM-Transmittermodule und RTC-Uhrenmodule wurde um weitere Module erweitert. Weitere Details entnehmen Sie bitte den technischen Daten.

7.1 Technische Daten

Die technischen Daten der Hardware Rev. 3.00 sind nachfolgend thematisch gegliedert aufgeführt:

Allgemeine Daten

Benennung	Wert	Einheit	Bemerkung
Betriebsspannung Logikteil	5	V	bestückt
Platinenabmessung	100x85	mm	
Kabelquerschnitt Klemmen	0,2-1,5	mm ²	
Gewicht	92	g	
Strombegrenzung Logikteil	ja, (Polyfuse)		

Digitale Eingänge

Benennung	Wert	Einheit	Bemerkung
Anzahl	8		
Art	Optokoppler		
Eingangsspannungsarten	AC/DC		
Eingangsspannung	24	V	
Überspannungstoleranz	36	V	für 10s
Verpolungsschutz	ja		
ESD Schutz	RC-Glied, Zenerdiode, Optokoppler		externe Absicherung notwendig
Strombegrenzung	nein		
Potentialtrennung	ja, (8x1)		
Logik	negativ		
EingangsfILTER	ja		
Eingangsspannung DC	4,4..24,0	V	High Pegel
Eingangsstrom DC	0..3,0	V	Low Pegel
	0,8..9	mA	High Pegel
	<=0,5	mA	Low Pegel
f_{max}	10	Hz	$U_E \geq 5V$
Eingangsspannung AC	9,0..24,0	V	High Pegel
Eingangsstrom AC	0..3,5	V	Low Pegel
	1,4..8,0	mA	High Pegel
	<=0,4mA	mA	Low Pegel
f_{max}	10	Hz	$U_E \geq 9V$

Digitale Ausgänge

Benennung	Wert	Einheit	Bemerkung
Anzahl	8		
Art	Transistor (offener Kollektor)		
Ausgangsspannungsarten	DC		
Betriebsspannung	5,0..24,0	V	
Überspannungstoleranz	30	V	
Verpolungsschutz	teilweise (nicht für ULN2803)		
ESD Schutz	ULN2803, Optokoppler		
Strombegrenzung	ja, (Polyfuse)		
Potentialtrennung	ja, (1x8 gegen andere Schaltungsteile)		
Logik	positiv		
Ausgangsleistung			bei 24V
I_{max} pro Kanal	100	mA	
I_{max} alle Kanäle	800	mA	

Sonstiges

Benennung	Wert	Einheit	Bemerkung
Schnittstellen	I2C Bus (kaskadierbar) 1wire 40 pol. Pfostenbuchse		auf insgesamt 8 Module auf Pin 7, pegelgewandelt auf 5V kompatibel zu diversen Kleincomputern
Konfiguration	MCP23017 INT-B I2C (Adressen 0x20..0x27)		Lötjumper auf Pin 13 oder 15 Steckjumper
Zusatzoptionen	RTC FM-Transmitter		verschiedene DS3231 Module STX882, MX-FS-03V

Anmerkungen

AC-Spannungen und Ströme sind, sofern nicht anders angegeben, Effektivwerte und beziehen sich auf eine Frequenz von 50Hz.

Die Spezifikation bezieht sich auf sachgemäß zusammengebaute Module unter Einsatz der von uns gelieferten Bauteile.

7.2 Stückliste

Die Stückliste des Hubo-Opto (Hardware Rev. 3.00) lautet wie folgt:

- 1 x Hubo Master-Platine (100x85; 35 μ Cu; doppelseitig durchkontaktiert, Lötstopplack, verzinkt)
- 35 x Anschlussklemmen (2- und 3-polig)
- 1 x MCP23017 (IO Expander IC1)
- 1 x ULN2803 (Treiber IC2)
- 1 x MOSFET 2N7000 (T100)
- 1 x Wannenstecker 2x13 polig (JP6)
- 2 x Stiftleiste 3 polig (JP1A, JP1B)
- 3 x Jumper
- 4 x IC Sockel 16 polig (Optokoppler)
- 1 x IC Sockel 18 polig (IC2)
- 1 x IC Sockel 28 polig (IC1)
- 3 x Kondensator 100nF (C101, C102, C103)
- 9 x Kondensator 22 μ F (C1..C8, C104)
- 1 x Kondensator 100 μ F-220 μ F (C100)
- 4 x Widerstand 4,7k Ω m (R1..R3, R100)
- 16 x Optokoppler (Single-Channel, bzw. Multichannel oder Mix)
- 9 x Diode 1N4148 oder äquivalent (D1..D8, D102)
- 2 x Widerstandsnetzwerk SIL8 4,7k Ω m (RN7, RN8)
- 6 x Widerstandsnetzwerk SIL8 1,0k Ω m (RN1..RN6)
- 2 x Widerstandsnetzwerk SIL8 1,8..2,2k Ω m (RN10, RN11)
- 1 x Widerstandsnetzwerk SIL9 4,7k Ω m (RN9)
- 2 x Schottky Diode BAT42 (D103, D104)
- 9 x Zenerdiode 1N4735A (D9..D16, D100)
- 2 x Polyfuse Sicherung (F1, F2)
- 1 x Widerstand 100 Ω m (R101)
- 4 x Abstandhalter für Hutschienegehäuse
- 4 x Mutter M3
- 4 x Schraube M3x12

Achtung, bei den gelieferten Optokopplern handelt es sich entweder um Single-Optokoppler oder 4-fach Optokoppler. Im Falle einer Mischbestückung von Single- und 4-fach Optokopplern sind die Single-Optokoppler an den Eingängen zu favorisieren. Die I2C-Pullupwiderstände R4 und R5 sind nicht Bestandteil des Bastelbeutels und sollten i.d.R. auch nicht bestückt werden, da die meisten Kleincomputer (z.B. alle derzeit unterstützten Raspberry Pi Modelle) bereits über integrierte Pullup Widerstände verfügen.

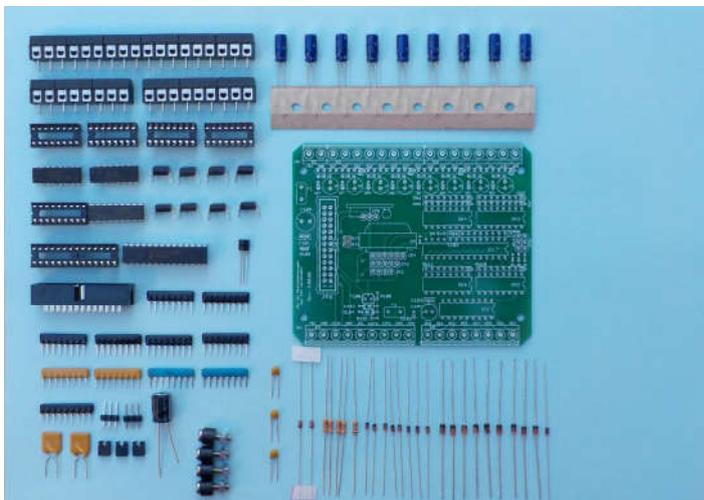


Bild 37: Bauteile des Hubo-Opto Hardware Rev. 3.00

7.3 Schaltplan

Den Schaltplan des Hubo-Opto Rev. 3.00 zeigt das folgende Bild 38.
Sehen sie auch dazu die Datei *Hubo_Schaltplan_3.00.00.pdf*.

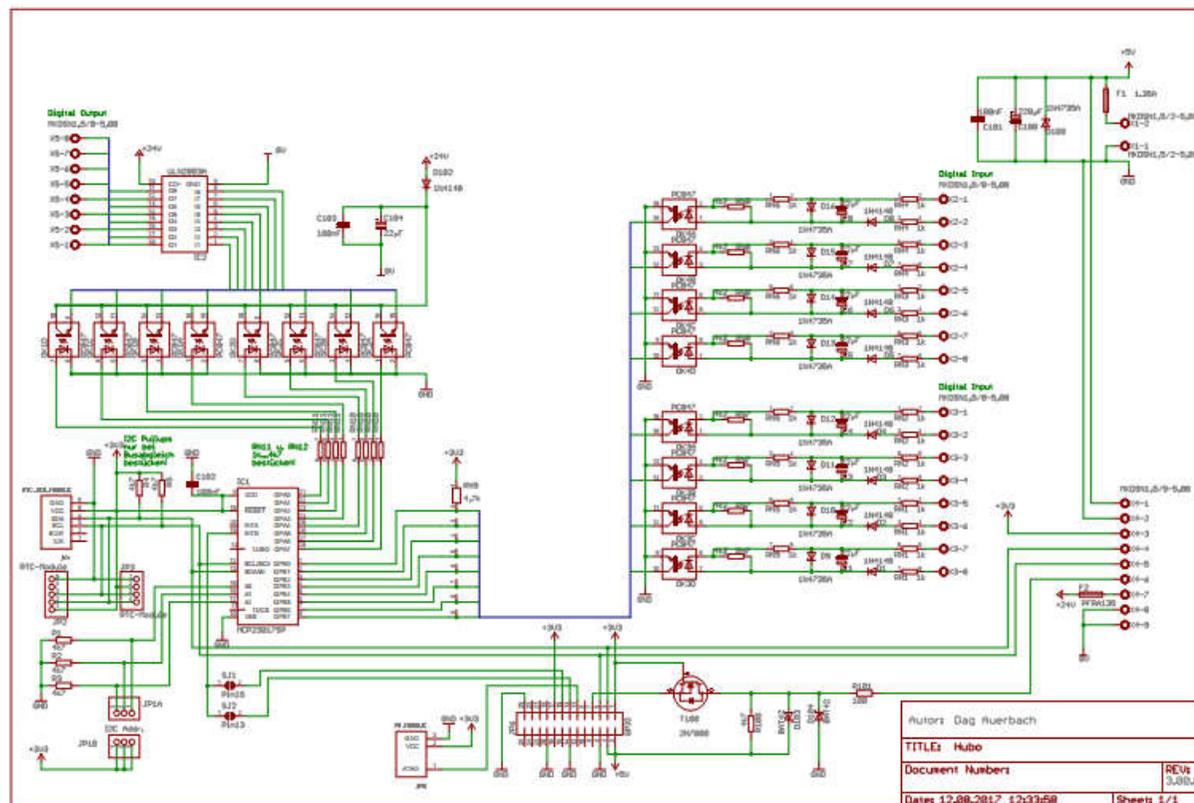


Bild 38: Schaltplan des Hubo-Opto (Hardware Rev. 3.00)

7.4 Arbeitsreihenfolge während der Bestückung

Die Bestückung des Hubo 3 erfolgt ähnlich der Bestückung der anderen Hubos. Zunächst werden die flachen Bauteile (100nF Kondensatoren, Schottky-Dioden, IC-Sockel, Widerstandsnetzwerke, ...) eingelötet, zuletzt die hohen Bauteile (... , stehende Dioden und Widerstände, Schraubklemmen, Elektrolytkondensatoren, Polyfuse-Sicherungen). Damit kann die Platine während des Lötvorgangs kopfüber positioniert werden, womit ein Durchrutschen der Bauteile während des Lötens verhindert wird. Bild 39 zeigt die vollständig bestückte Platine der Hardware Rev. 3.00.

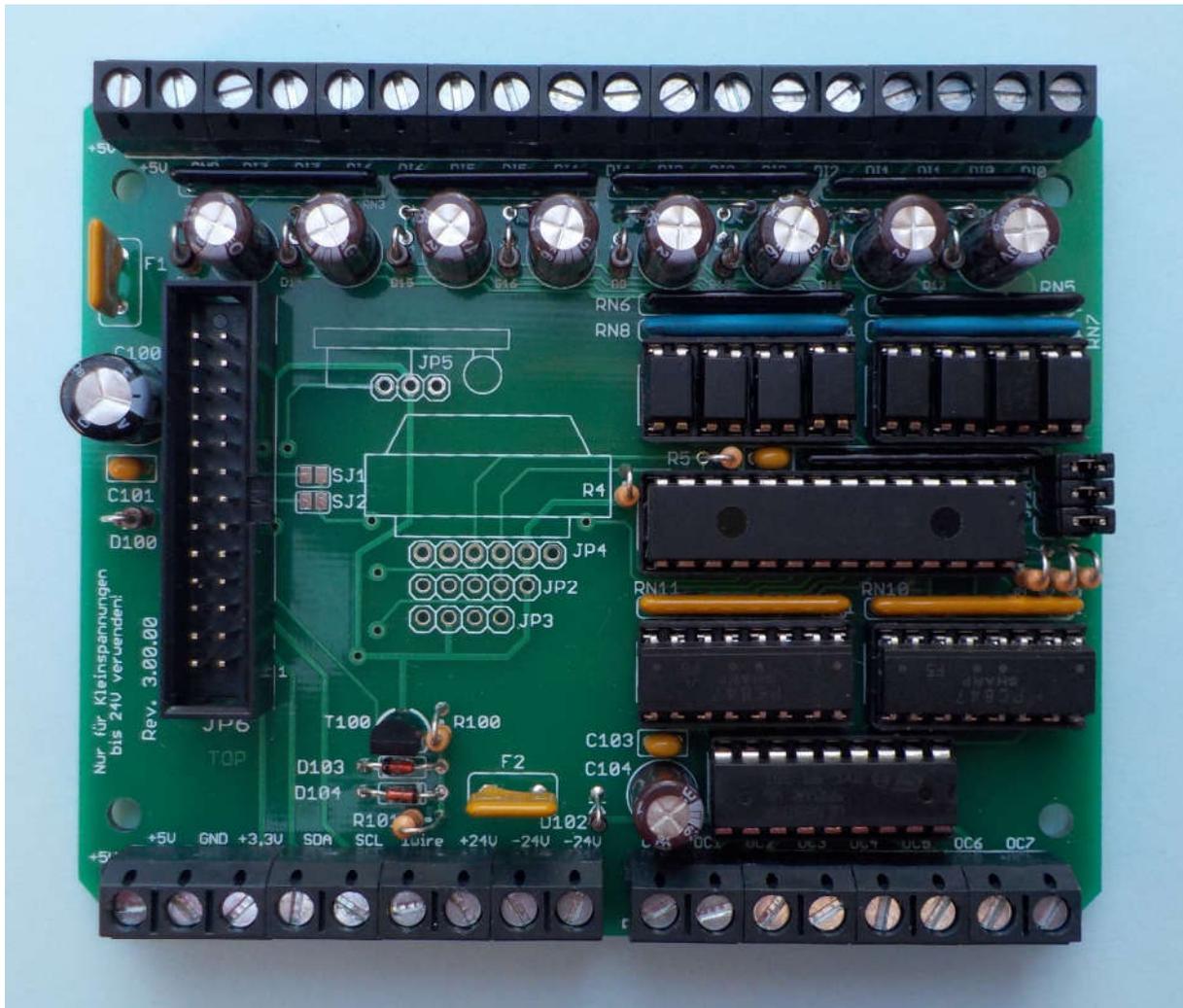


Bild 39: Hubo-Opto Hardware Rev. 3.00 nach der Bestückung

Beachten Sie bitte, daß die im Bild 38 bestückten I2C Pullup-Widerstände R4 und R5 nicht Bestandteil des Bausatzes sind und i.d.R. auch nicht bestückt werden sollten. Sollte INTB des MCP23017 auf einen der GPIO's (Wannenpfostenstecker Pin 15 bzw. Pin 13) gelegt werden, so sind die Lötjumper SJ1 bzw. SJ2 entsprechend zu überbrücken.

8 Hardwarerevisionen

Die Varianten des Hubo werden von uns kontinuierlich weiterentwickelt und verbessert. Die derzeit aktuellen Hardwarerevisionen lauten Rev. 1.00.03, Rev. 1.11.00, Rev. 2.01.00 und Rev. 3.00.00.

Abweichungen in der rechten Zifferngruppe kennzeichnen nur geringfügige Änderungen innerhalb der Produktgruppe (z.B. geänderter Bestückungsaufdruck). Abweichungen in der mittleren Zifferngruppe deuten auf größere Änderungen z.B. der Funktionalität oder des Layouts hin. Eine Änderung der vorderen Ziffer deutet auf einen grundsätzlichen Technologieunterschied hin.

9 Inbetriebnahme der Hardware und Programmierung

Sofern nicht anders vermerkt, gelten die Beispielbeschaltungen für alle Hardware Revisionen.

9.1 Stromversorgung von Hubo und Raspberry Pi

Die Stromversorgung des Raspberry Pi kann auf drei unterschiedlichen Wegen erfolgen:

- über die dafür vorgesehenen Micro-USB Buchse,
- die USB-Anschlüsse bzw.
- über die GPIO Pfostenleiste.

Da der Hubo ebenfalls eine 5V Stromversorgung benötigt, stellt sich die Frage, wie beide Geräte am sinnvollsten versorgt werden. Nachdem der Hubo bei angezogenen Relais (bei angeschlossener Relaisoption können das insgesamt 12 Relais sein) einen höheren Strombedarf als der Raspberry Pi hat, ist es sinnvoll, die Einspeisung am Hubo (Buchse X1) vorzunehmen. Dieser leitet dann die notwendige 5V Versorgungsspannung über das Flachbandkabel (JP4) an den Raspberry Pi weiter (siehe Bild 40).

Achtung! Achten Sie beim Anschluß der 5V Versorgungsspannung an den Hubo auf die richtige Polung! Ein fehlerhafter Anschluß kann eine Zerstörung des Hubo bzw. auch der des angeschlossenen Raspberry Pi zur Folge haben.

An X1-1 wird dabei GND (der Minuspol) und an X1-2 die 5V (Pluspol) angeschlossen.

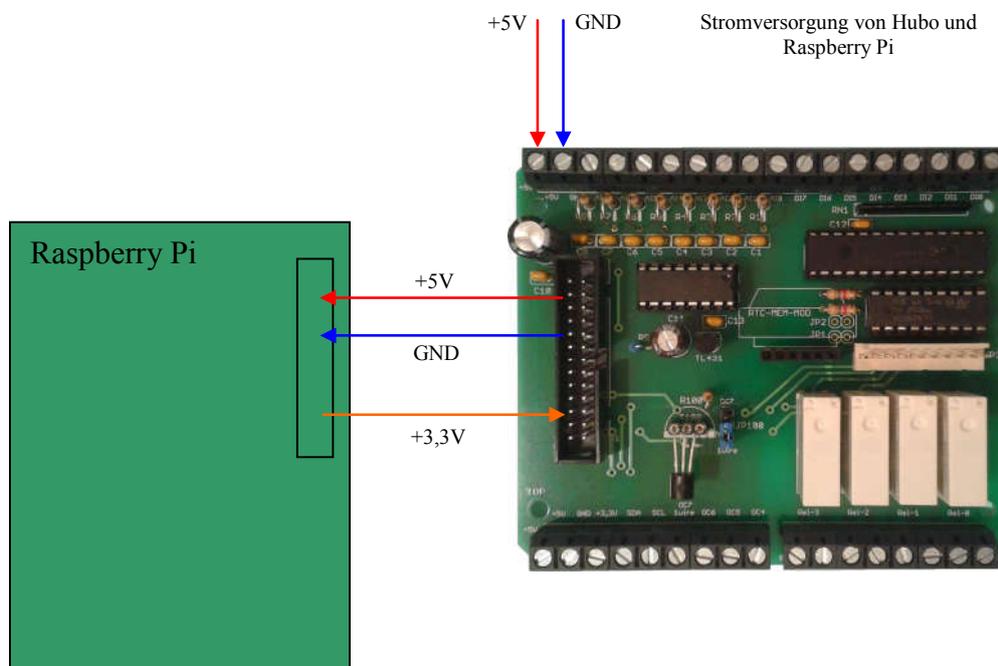


Bild 40: 5V Stromversorgung des Raspberry Pi über den Hubo; Rückspeisung der 3,3V in den Hubo.

Bei der Konfektionierung des Flachbandkabels ist darauf zu achten, dass Pin 1 des Hubo (JP4) mit Pin 1 der Pfostenleiste der Raspberry Pi verbunden wird. Zur entsprechenden Markierung sollte die rote Ader des Flachbandkabels jeweils an Pin1 der entsprechenden Pfostenstecker gequetscht werden.

9.2 Kaskadierung mehrerer Hubos

Durch Hintereinanderschaltung (Kaskadierung) mehrerer Hubos können dessen digitale Ein- und Ausgänge auf bis zu 32 Eingänge und 31+1 Ausgänge erweitert werden. Dazu wird der I2C Bus vom Master Hubo zu den bis zu 3 weiteren Slave Hubos „geschlaucht“. Dies kann entweder per Flachbandkabel erfolgen (alle Hardware Rev.), idealerweise jedoch über die diesbezüglichen Schraubklemmen (ab Hardware Rev. 1.11).

Zur Führung des Busses empfiehlt sich dabei ein geschirmtes Kabel, dessen Schirm am Master Hubo an GND angekoppelt wird. Die Schirmung sollte zur Vermeidung von Erdungsschleifen **nicht** an den Slaves angekoppelt werden.

Achten sie bei der Stromverteilung vom Netzteil zu den Hubos darauf, daß sie alle Hubos (Master und Slaves) sternförmig vom Netzteil mit 5V Versorgungsspannung versorgen. Sie vermeiden somit Rückwirkungen von Spannungsschwankungen durch schaltende Relais.

Achtung! Beachten Sie, dass die Hubo Slaves eine eindeutige I2C Bus Adresse benötigen. Diese wird mittels der Jumper JP1 und JP2 eingestellt. Der Master Hubo wird nicht ge-jumpert und bleibt damit auf Adresse 0x20.

Hinweis! Da der analoge Teil eines Hubos nicht kaskadierbar ist, stellt der Hubo-Digital (Hardware Rev. 1.11) die ideale Ausgangsbasis für kaskadierte Hubos dar.

9.3 Referenzspannungserzeugung

Die zur analogen Messung notwendige Referenzspannung wird aus der Referenzspannungsquelle TL431 auf dem Hubo erzeugt. Um keine zu hohen Anforderungen an die Stabilität und Glättung der 5V Speisespannung der Hauptstromquelle zu stellen, wird die für die Referenzspannungsquelle notwendige Eingangsspannung nicht aus den 5V der Speisespannung des Hubo bezogen, sondern aus den 3,3V, welche auf dem Raspberry Pi erzeugt werden (siehe dazu die orange Linie in Bild 32). Die Referenzspannungsquelle erzeugt daraus eine zwischen 2,44V und 2,55V (typisch 2,495V) liegende konstante Spannung. Es wird empfohlen, diese Spannung nachzumessen und diesen Referenzwert bei analogen Messungen mit zu berücksichtigen. Im Falle der Verwendung der Hubo C++-Library erfolgt das über einen Aufruf der Funktion `HuboLib::Set_VRef(double volt)`.

9.4 Anschluß digitaler Sensoren

Hubo verwendet den MCP23017 IO Expander zum Einlesen und Ausgeben digitaler Signale. Da die im IO-Expander integrierten und zuschaltbaren Pullupwiderstände mit einem Wert von 100k sehr hoch dimensioniert sind, eignen sie sich i.d.R. allein nicht, um Schaltsignale weit entfernt liegender Schalter störungsarm zu erfassen. Über lange Kabel einkoppelnde Störsignale verfälschen bei zu hochohmig ausgelegten Eingängen das einzulesende Signal. Um den Eingangswiderstand den Bedürfnissen vor Ort anzupassen, besitzt Hubo ein Widerstandnetzwerk (RN1), welches zwischen 470Ohm bis etwa 10k bestückt werden kann. Damit ergibt sich eine bessere Unterdrückung eingekoppelter Störungen. Wie bereits beschrieben, handelt es sich um Pullupwiderstände, womit (im Normalfall) die Kanäle softwareseitig als „ON“ erkannt werden. Um eine entsprechende Schaltwirkung zu erzielen, müssen angeschlossene Schalter die digitalen Eingänge somit gegen Masse ziehen. D.h. ein externer Schalter verbindet einen der digitalen Eingänge DI0 (X2-1) bis DI7 (X2-8) mit dem Masseanschluß an Klemme X5-2.

Achtung! Beachten Sie, dass die digitalen und analogen Eingänge von rechts nach links nummeriert sind.

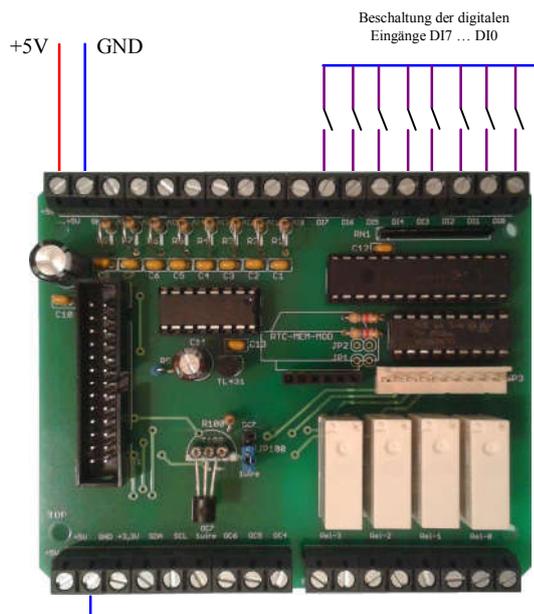


Bild 41: Beschriftung digitaler Eingänge gegen Masse.

9.5 Analoge Spannungsmessung

Spannungen können bis zu einem Bereich der Referenzspannungsquelle (also etwa 2,5V) direkt an den analogen Eingangsklemmen AI0 (X3-1) bis AI7 (X3-8) gemessen werden (siehe Bild 42). Sollen höhere Spannungen als 2,5V gemessen werden, so muß ein entsprechender Spannungsteiler vorgeschaltet werden. Wie bei der digitalen Signaleingabe erfolgt die Messung zwischen einem Eingang und dem Masseanschluß an Klemme X5-2.

Achtung! Beachten Sie, dass die digitalen und analogen Eingänge von rechts nach links nummeriert sind.

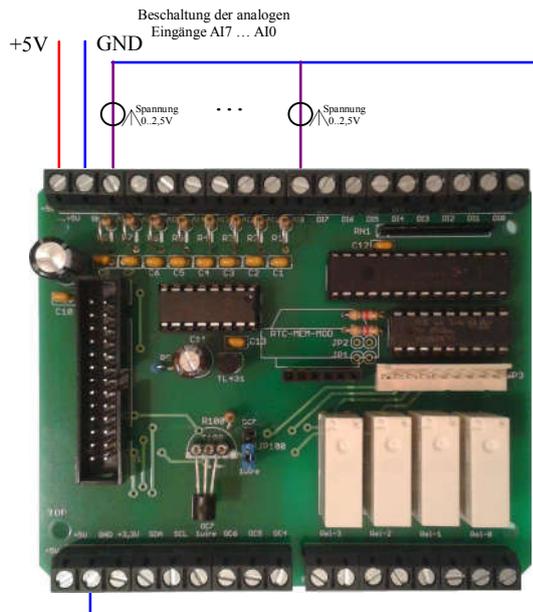


Bild 42: Messung analoger Spannungen zwischen 0..2,5V.

Achtung! Im Falle des Hubo-Maxi Hardware Rev. 2.01 entfällt der Analogeingang AI7 (X3-8). Der Eingang X3-8 stellt den analogen Ground zur Analogspannungsmessung dar (Bild 43).

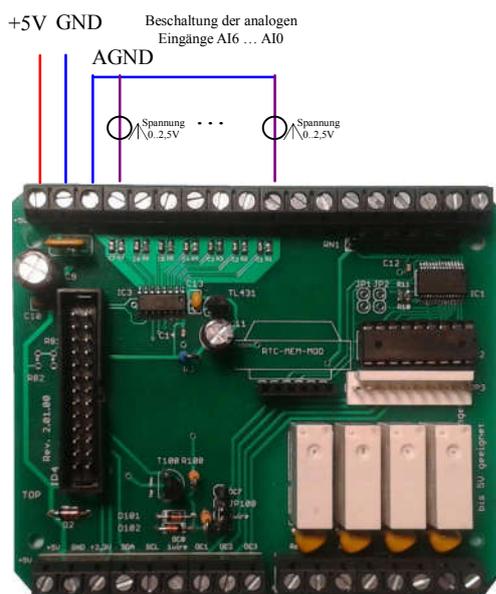


Bild 43: Analoge Spannungsmessung beim Hubo-Maxi Hardware Rev. 2.01.

9.6 Temperaturmessung mittels MCP9700/MCP9701

Die analogen Temperatursensoren MCP9700 und 9701 liefern ein der Temperatur proportionales Signal zwischen 0 und etwa 2,5V bei den gegebenen Temperaturbereichen von -40°C bis 125°C. Das gemessene Signal wird im Chip entsprechend vorverstärkt und besitzt eine Linearität, die für viele Anwendungen lediglich einer Einpunktkalibrierung bedarf. Die Stromversorgung der Sensoren kann für beide Varianten zwischen 3,2V und 5,5V liegen. Idealerweise entscheidet man sich für die am Hubo an den Klemmen X5-1 (+5V) und X5-2 (GND) zur Verfügung stehenden 5V. Das Ausgangssignal des MCP970x wird direkt an einen der analogen Meßeingänge X3-1 bis X3-8 des Hubo angelegt (siehe Bild 44). Die Hubo C++-Library bietet Funktionen zur direkten Konvertierung der gemessenen Spannung in die dazugehörige Temperatur (in °C).

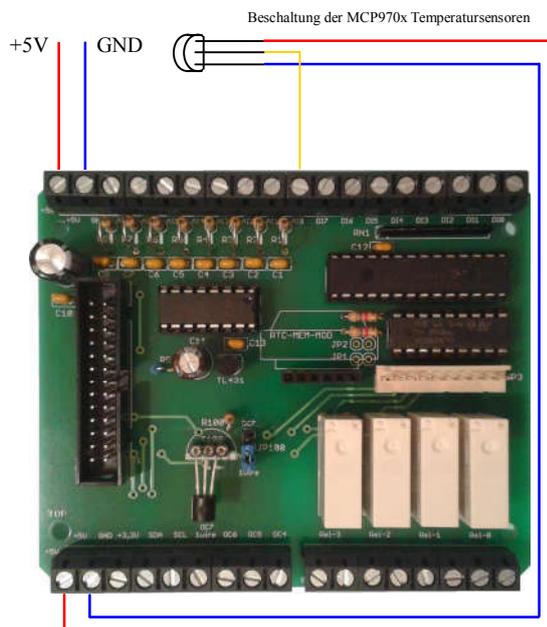


Bild 44: Analoge Temperaturmessung mittels der Sensoren MCP970x (hier am analogen Kanal 0).

9.7 Anschluß von 1wire Modulen

Der Name „1wire“ ist eigentlich irreführend, da zum Betrieb der Sensoren mindestens 2 Drähte benötigt werden. Der Massedraht GND wird generell benötigt. Beim normalen Betrieb werden die Versorgungsspannung und die Signalleitung auf jeweils getrennten Kabeln geführt, womit sich in der Summe 3 Leitungen ergeben. Allerdings lassen sich Versorgungsleitung und Signalleitung auch auf ein und derselben Leitung realisieren, womit ein 2-adriges Kabel für derartige Devices ausreichend ist. Auf die Besonderheiten bei der Busauslegung wird weiter hinten nochmals ausführlich eingegangen. An dieser Stelle soll nur kurz die Beschaltung am Hubo erläutert werden.

Achtung! Stellen Sie in jedem Fall sicher, dass sich der Jumper JP100 in der Stellung „1wire“ befindet. Nur in diesem Fall ist eine Kommunikation zwischen Hubo und 1wire Device möglich. Das gilt sowohl für den Normalbetrieb, wie auch den parasitären Betrieb von 1wire Devices.

9.7.1 Normalbetrieb an 3 Leitungen

Beim 3-Draht Betrieb verbindet man die 5V Versorgungsspannung von Klemme X5-1 (+5V) mit der Versorgungsleitung des 1wire Devices, analog dazu verbindet man die Masseleitung an Klemme X5-2 (GND) des Hubo mit dem Masseanschluß des 1wire Devices. Die 1wire-Datenleitung des Hubo auf Klemme X5-6 wird mit der entsprechenden Datenleitung (in ocker dargestellt) des 1wire-Slaves verbunden (siehe Bild 45).

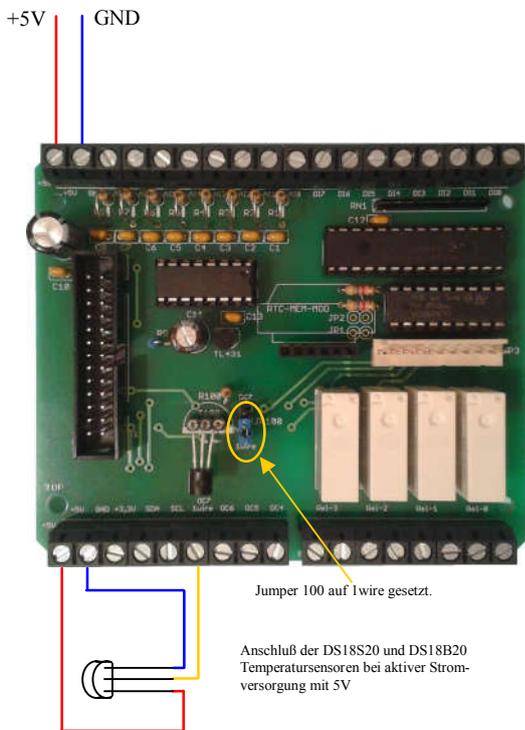


Bild 45: 1wire Betrieb bei aktiver Stromversorgung des Devices von 5V

9.7.2 Parasitärer Betrieb mit 2 Leitungen

Der Anschluß von Daten- und Masseleitung zwischen Hubo und 1wire Device erfolgt wie bei beim Normalbetrieb mit 3 Leitungen. Für die meisten 1wire Slaves ist danach allerdings am 1wire Device selbst die Versorgungsspannung auf den Masseanschluß zu brücken (siehe Bild 46). Dies signalisiert dem Slavedevice, dass die Spannungsversorgung parasitär aus der Datenleitung bezogen wird. Für nähere Informationen dazu konsultieren Sie bitte das Datenblatt des 1wire Devices.

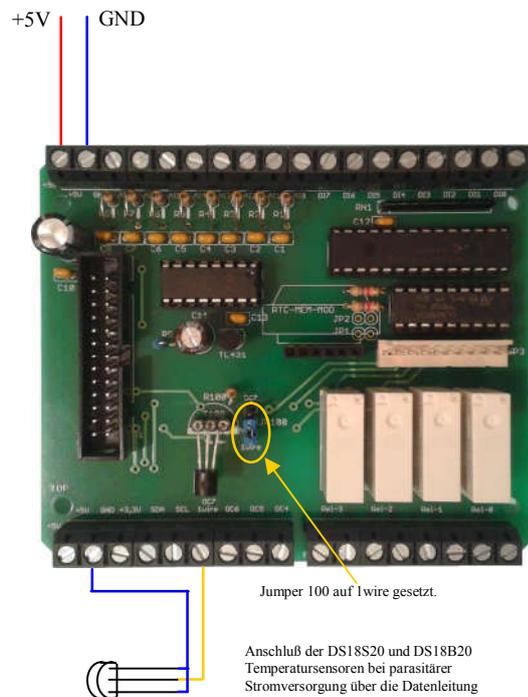


Bild 46: 1wire Betrieb bei parasitärer Stromversorgung des Devices

9.7.3 Temperaturmessung mittels DS18x20

Verbinden Sie die Temperatursensoren auf eine der oben beschriebenen Art und Weise. Sind alle Kerneltreiber geladen, so sollten Sie jetzt unter `/sys/bus/w1/devices` eine Liste erkannter 1wire Devices einschließlich des Busmasters finden. Die Temperatursensoren besitzen eine entweder mit 28 oder 10 beginnende Geräteklassen-ID. Wechseln Sie in eines der Verzeichnisse und führen Sie `cat w1_slave` aus. Bei ordnungsgemäßer Funktionsweise sollten Sie jetzt ungefähr folgende die Ausgabe erhalten:

```
5c 01 4b 46 7f ff 04 10 a1 : crc=a1 YES
5c 01 4b 46 7f ff 04 10 a1 t=21750
```

Weiterführende Hinweise zum Test mittels der Demoprogramme der Hubo C++-Library finden Sie in den einschlägigen Kapiteln.

9.8 Die Programmierung mittels der Hubo C++-Library

Die Hubo C++-Library wurde ausdrücklich zur Unterstützung der Hubo Hardware entwickelt. Mit Hilfe dieser Bibliothek testen wir auch unsere Entwicklungen, vor einer Vermarktung.

Die Hubo C++-Library enthält etliche vorkompilierter Beispiele, die den Einsatz der Bibliotheksfunktionen verdeutlichen bzw. bei der Inbetriebnahme der Hardware nützlich sind. Zur Installation der Bibliothek und der notwendigen Kernelmodule enthält die Hubo C++-Library ein Installationscript.

Mittels RT scheduling Policy und Hintergrundbearbeitung erlaubt die Bibliothek (im Rahmen der Möglichkeiten des Betriebssystems) eine zeitlich sehr genaue Messwertübertragung und birgt für die Datenintegrität sämtlicher Messwerte innerhalb eines Taktzyklus. Events erlauben eine taktsynchrone Datenverarbeitung durch die Applikation. Der Zyklentakt kann parametrisiert werden und reicht von 100Hz (beim Einlesen aller Analogeingänge – ggf. mit Oversampling) über 500Hz (digitale Eingänge und ein einzelner Analogkanal) bis hin zu 1kHz (reines Einlesen der digitalen Eingänge).

9.8.1 Lesen und Schreiben der digitalen Kanäle

Das folgende Beispiel zeigt, wie die digitalen Kanäle des Hubo mittels der Hubo C++-Library gelesen und geschrieben werden können.

```
#include <stdio.h>
#include <unistd.h>
#include "../hubolib.h"

using namespace HuboLib;

int main(void)
{
    unsigned char channel = 7;
    bool bState = false;

    // Initialize the library once in your program.
    Initialize();
    usleep (1000L*10);

    // Read digital input.
    Get_DI_Channel(channel, bState);
    printf ("Digital input %d = %s \n", channel, bState ? "ON " : "OFF");

    // Write value to digital output.
    Set_DO_Channel(channel, bState);

    // Free library resources.
    usleep (1000L*10);
    Uninitialize();

    return 0;
}
```

9.8.2 Lesen der analogen Kanäle

Das folgende Beispiel zeigt, wie die analogen Kanäle des Hubo mittels der Hubo C++-Library gelesen werden können.

```
#include <stdio.h>
#include <unistd.h>
#include "../hubolib.h"

using namespace HuboLib;

int main(void)
{
    // Initialize the library once in your program.
    Initialize();

    // Define channel 2 to be used as source of input.
    unsigned short overSampling[MAX_MCP3x08_CHANNELS] = { 0, 0, 1, 0, 0, 0, 0, 0 };
    Set_MCP3x08_Oversampling (overSampling);

    int          channel = 2;
    unsigned long adcCount;
    double       volt;
    for (int i=0; i<20; i++)
    {
        // Read analog channel.
        Get_AI_Channel (channel, adcCount, volt);
        printf ("Channel=%d  ADC count=0x%02lX  Volts=%lf\n", channel, adcCount, volt);

        usleep(1000L*1000L);
    }

    // Free library resources.
    Uninitialize();

    return 0;
}
```

9.8.3 Inbetriebnahme mittels der Hubo C++-Library Demoprogramme

Nach dem erfolgten Zusammenbau kann die korrekte Funktionsweise der Hubo Hardware am einfachsten mittels der Demoprogramme der Hubo C++-Library überprüft werden.

Es sei an dieser Stelle daher auf die folgenden Beispiele verwiesen. Beachten Sie bitte, dass nahezu alle Beispiele mit Superuser-Rechten (sudo) ausgeführt werden müssen.

LoadTest

Dieses Beispiel ermöglicht das Setzen einer oder mehrerer digitaler Ausgänge Hubos. So schaltet *sudo LoadTest.out 255* alle Ausgänge ein, *sudo LoadTest.out 0* löscht hingegen alle Ausgänge. Gleichzeitig wird ein Core der CPU in eine Endlosschleife gezwungen. Das Programm muß per STRG-C beendet werden.

DigitalInput1

Nach dem Start per *sudo DigitalInput1.out* werden die Digitalkanäle 0 und 1 zyklisch gepollt und deren Status angezeigt. Nachdem Hubo Pullupwiderstände gegen Hi verwendet, sind die Kanäle standardmäßig eingeschaltet. Eine Beschaltung gegen Masse entspricht daher der normalen Verwendung und führt zur Signalisierung des Lo-Zustandes. Beenden Sie das Programm mittels STRG-C.

Achtung: Beachten Sie, dass die Zählreihenfolge der Eingänge von rechts nach links erfolgt!

AnalogInput1

sudo AnalogInput1.out gibt die analogen Werte, gemessen am analogen Eingang 0 aus. Gemäß der Implementierung des Beispiels ist es korrekt, dass die ersten Ausgaben eine Spannung von 0 darstellen und erst nachfolgende Ausgaben den angelegten Spannungswert darstellen.

Als Prüfmittel kann hier eine 1,5V Batterie, ein 1,2V Akku o.ä. verwendet werden.

Achtung: Beachten Sie, dass die Zählreihenfolge der Eingänge von rechts nach links erfolgt!

DS18x20

Mit Hilfe des Beispiels *DS18x20.out* werden alle am 1wire Bus angeschlossenen DS18x20 aufgelistet und deren gemessene Temperaturwerte ausgegeben. Dieses Programm benötigt keine su-Rechte.

DigitalSlaveOutput1

Das Beispiel *DigitalSlaveOutput1.out* dient der Überprüfung der korrekten Funktionsweise des auf Index 0 kaskadierten Hubo-Slavemodules (typischerweise liegt dies auf I2C Adresse 0x21). Starten Sie dieses Programm mittels *sudo DigitalSlaveOutput1.out*.

DigitalSlaveInput1

Das Programm *DigitalSlaveInput1.out* verhält sich analog zu *DigitalInput1.out*, wobei jedoch die digitalen Eingänge 0 und 1 des auf Index 0 kaskadierten Hubo-Slavemoduls gepollt werden. Starten Sie dieses Programm mittels *sudo DigitalSlaveInput1.out*, beenden Sie es mittels STRG-C.

Weitere Informationen zur Verwendung und Installation können Sie der Dokumentation (in Englisch) der Hubo C++-Library entnehmen.

9.8.4 Generelle Anmerkungen zur Verwendung der Hubo C++-Library

Beachten Sie jedoch, dass die Hubo C++-Library eine kostenfreie Software ist, die nicht Gegenstand dieses Bausatzes ist.

Wenden Sie sich im Bedarfsfall an uns, um eine Kopie der Software zu erhalten.

9.9 Die Programmierung in Python und anderen Sprachen

Grundsätzlich können alle Hubo Boards auch in anderen gängigen Sprachen (wie z.B. Python unter Einsatz der Bibliotheken `py-spidev` und `python-smbus`) betrieben werden. Anbei finden sich zwei einfache Beispiele, die das Lesen und Schreiben digitaler und analoger Daten verdeutlichen.

9.9.1 Lesen und Schreiben der digitalen Kanäle in Python

Das folgende Beispiel zeigt, wie die digitalen Eingänge des Hubo gelesen werden bzw. wie die Ausgänge beschrieben werden können.

```
#!/usr/bin/python

import smbus
import time

# Hubo MCP23017 Demo - simple reading and writing of the MCP23017 digital IO.

IODIRA = 0x00
IODIRB = 0x01
GPPUB = 0x0D
GPIOA = 0x12
GPIOB = 0x13

global g_ioaddress
g_ioaddress = 0x20 # I2C address - Hubo master modules uses 0x20

def Initialize_MCP23017 ():
    i2cbus = smbus.SMBus(1) # Use 1 for the model B variants.
    i2cbus.write_byte_data (g_ioaddress, IODIRA, 0x00) # Configure port A as output
    i2cbus.write_byte_data (g_ioaddress, IODIRB, 0xFF) # Configure port B as input
    i2cbus.write_byte_data (g_ioaddress, GPPUB, 0xFF) # Activate port B's internal pull up resistors
    return i2cbus

def Get_DI_Channels (i2cbus):
    return i2cbus.read_byte_data(g_ioaddress, GPIOB)

def Set_DI_Channels (i2cbus, value):
    return i2cbus.write_byte_data(g_ioaddress, GPIOA, value)

# Read values from the digital input port and output the value on the digital output port.
i2cbus = Initialize_MCP23017()
while True:
    value = Get_DI_Channels (i2cbus)
    Set_DI_Channels(i2cbus, value)
    print value
    time.sleep(0.1)
```

9.9.2 Lesen der analogen Kanäle in Python

Das folgende Beispiel zeigt, wie die analogen Eingänge des Hubo gelesen werden können.

```
#!/usr/bin/python

import spidev
import time

# Hubo MCP3208 Demo - simple reading of the MCP3208 analog input.

def Open_SPI_ADC ():
    spiADC = spidev.SpiDev()
    spiADC.open(0, 0)
    spiADC.max_speed_hz = (50000)
    return spiADC

def Get_AI_Channel (spiADC, channel):
    if ((channel > 7) or (channel < 0)):
        print 'Illegal channel number (0 through 7)'
        return 0.0
    result = spiADC.xfer2([6 + (channel >> 2), channel << 6, 0])
    count = ((result[1] & 0x0F) << 8) + (result[2])
    # Use a reference voltage of 2.53V.
    voltage = (2.53 * count) / 4096
    return voltage

# Read 10 values from analog input 0.
spiADC = Open_SPI_ADC()
for i in range(0, 9):
    print Get_AI_Channel (spiADC, 0)
    time.sleep(1)
spiADC.close();
```

10 Empfehlungen

10.1 Hardwareergänzungen der Hubo-Produktreihen der Hardwarerevisionen 1 und 2

10.1.1 Hutschienengehäuse

Wie bereits ausgeführt, wurden die Hubo Boards speziell zur Montage im Hutschienengehäuse konzipiert. Bild 47 zeigt den Einbau Hubo-Maxis im Gehäuseunterteil bei gleichzeitiger Einbettung des Raspberry Pi (möglich bei den Modellen Zero, A+, B+, 2 B, 3 B bzw. 3 B+) im Oberteil des Gehäuses. Beachten Sie, dass bei dieser Verwendung die RTC Option aufgrund der beschränkten Bauhöhe nicht mehr möglich ist. Stecken Sie in einem solchen Fall die RTC in ein kaskadiertes Slavemodul.



Bild 47: Hubo-Maxi Hardware Rev. 1.11 und Raspberry „Pi B+“ im Hutschienengehäuse

Beachten Sie, dass das ältere Raspberry Modell „Pi B“ nicht in den Deckel des Gehäuses passt. Weichen Sie dann auf ein spezielles Hutschienengehäuse für diesen Typ aus. Allerdings müssen Sie dann auch eine Flachbandkabeldurchführung in das Hutschienengehäuse einarbeiten.

10.1.2 Hutschienennetzteil

Zur Sicherstellung einer stabilen Stromversorgung des Raspberry Pi bei gleichzeitig schaltenden Relais, wird ein starkes Netzteil benötigt. Um den Formfaktor der Hutschiene zu ergänzen, wird ein ebensolches empfohlen. Vorteilhaft ist bei diesem Netzteil, dass die Spannung sich in gewissen Grenzen einstellen lässt. Die analogen Eingangsfiler des Hubo-Maxi Boards wurden darauf abgestimmt, Oberwellen dieses Netzteils zu filtern.

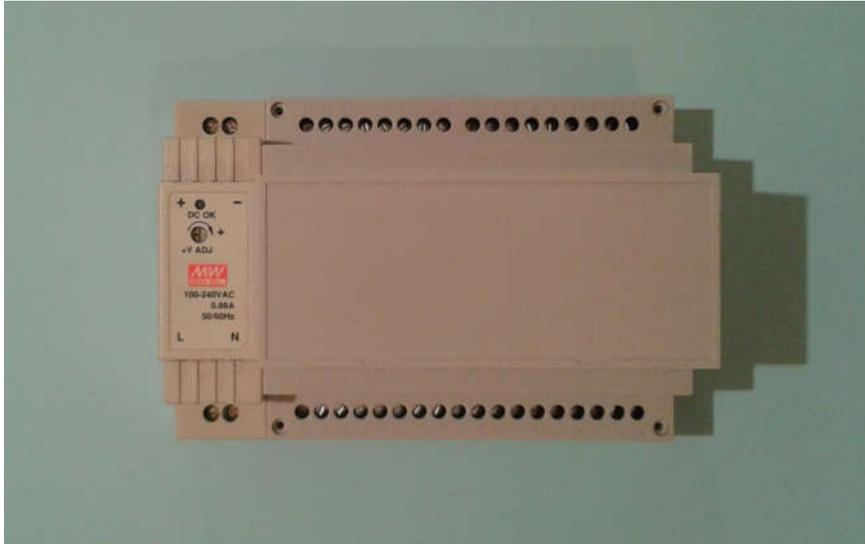


Bild 48: Hubo mit Spannungsversorgung aus dem Hutschienennetzteil

10.1.3 DS3231 Real Time Clock und 433MHz Sender

Mithilfe der DS3231 Uhr behält der Pi die Uhrzeit auch nach einem Stromausfall. Mittels des 433MHz Transmittermoduls können u.a. Funksteckdosen angesteuert werden.

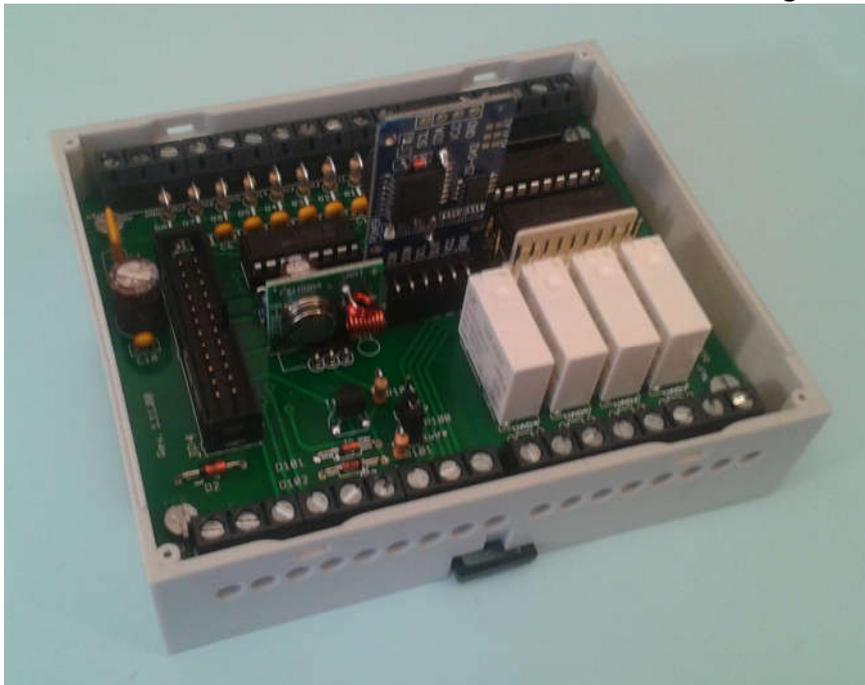


Bild 49: Hubo-Maxi Hardware Rev. 1.11 mit RTC- und 433MHz Sendemodul

10.1.4 Externes Lastrelais zum Schalten größerer Lasten

Sollen mit Hubo (gilt gleichermaßen für Hubo-Maxi und Hubo-Digital) größere Lasten geschaltet werden, so sollte ein dafür taugliches Lastrelais eingesetzt werden.



Bild 50: Hubo Hardware Rev. 1.00 mit externem Lastrelais (gilt für alle Hardware Rev.)

10.1.5 Alle Optionen im Überblick



Bild 51: Alle Hardwareoptionen im Überblick (Sender nur für Hardware Rev. 1.11)

10.2 Hardwareergänzungen zur Hubo-Produktreihe der Hardwarerevision 3

10.2.1 Hutschienengehäuse

Es gelten die gleichen Aussagen wie für die Produktreihen der Hardwarerevisionen 1 und 2.

10.2.2 Hutschienennetzteil

Bezüglich der 5V Stromversorgung des Raspberry Pi gelten die gleichen Aussagen wie für die Produktreihen der Hardwarerevisionen 1 und 2.

Um die galvanische Trennung der digitalen Ausgänge beizubehalten, sollte die Stromversorgung der daran angeschlossenen Aktoren über ein separates Netzteil erfolgen. Bild 52 zeigt ein mögliches 24V DC Netzteil, welches z.B. zur Versorgung von Koppelrelais Verwendung finden kann.



Bild 52: 24V Netzteil zur Speisung galvanisch getrennter Aktoren (z.B. Koppelrelais)

10.2.3 DS3231 Real Time Clock und 433MHz Sender

Die Hardwarerevision 3 erlaubt den Anschluß einer größeren Vielfalt von RTC- und FM-Transmittermodulen als die Vorgängerversionen. Bild 53 zeigt einige dieser zusätzlich unterstützten Module.

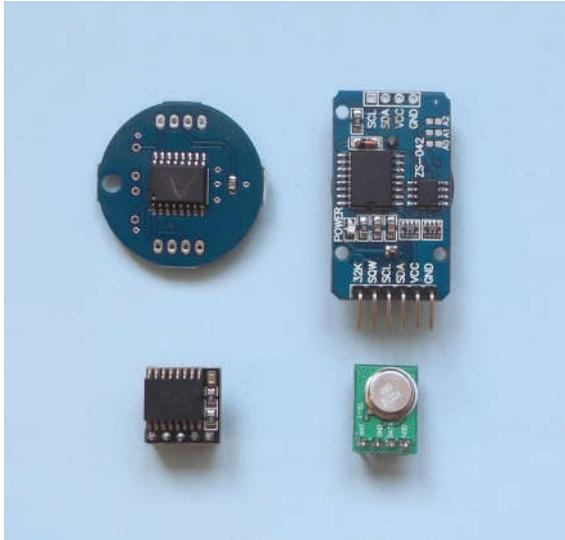


Bild 53: Zusätzliche unterstützte RTC- und 433MHz Sendemodule

10.3 1wire Schnittstelle

Wenngleich Hubo mit seinen 8 analogen Schnittstellen bereits eine Vielzahl analoger Sensoren direkt unterstützt, so ist die 1wire Schnittstelle dennoch eine bequeme Option, will man mit wenig Aufwand die beliebten DS18x20 Temperatursensoren betreiben. Beachten Sie dabei jedoch, dass die Verdrahtungstopologie eine sehr wichtige Rolle bei der Stabilität der Buskommunikation spielt. Es wird dringend empfohlen die Applikationshinweise der verschiedenen Hersteller im Vorfeld zu studieren. So stellen sternförmige Verkabelungen aufgrund der vielen offenen Endpunkte Quellen von Signalreflexionen dar. Idealerweise sollte eine Verkabelung nicht sternförmig erfolgen.

Da beim Design der Hubo Hardware nicht vorhergesehen werden konnte, welche Art der Verkabelung letztendlich zum Einsatz gelangt, wurden diese so ausgelegt, dass der Pullup-Widerstand des Boards mit 4,7kOhm dem empfohlenen Referenzwert der Applikationsnote des DS18x20 entspricht. Mit diesem Setup wurden 9 verschiedene Sensoren (B und S Typen) an einem einzelnen Bus der Länge mehrerer Meter betrieben.

Für den konkreten Einsatz kann es allerdings sinnvoll sein, den Widerstandswert zu erniedrigen. Das ist dann der Fall, wenn z.B. die Anzahl der Busteilnehmer größer ist, die Kabellänge wesentlich länger ausfällt oder andere Busteilnehmer einen größeren Strombedarf als die o.g. Temperatursensoren haben (z.B. 1wire Speicherchips). Als Faustregel kann der Wert bis auf etwa 1kOhm ohne weitere Bedenken abgesenkt werden. Werte darunter signalisieren i.d.R. eine schlechte Verkabelung (Echos, Übergangswiderstände, Signalstörungen). Verwenden Sie zur Inbetriebnahme der DS18x20 Sensoren das Programm DS18x20.out und beobachten Sie die Zeiten zur Ermittlung des Temperaturwertes. Weichen diese gravierend von 820ms ab, so sollten Sie Ihre Verkabelung überprüfen.

Nachdem Hubo lediglich eine Signalpegelanpassung von 3,3V auf 5V realisiert, ansonsten jedoch den Bitbang Mode des Linux Kernaltreibers verwendet (sich also kein separater 1wire Controller auf dem Board befindet), ist die Stabilität der Kommunikation des Treibers in gewissen Grenzen auch CPU lastabhängig. Wo die konkreten Grenzen dabei liegen, kann mit dem LoadTest-Demo der Hubo C++-Library herausgefunden werden. Setzt man den Zyklentakt der Bibliothek auf 5ms oder niedriger und verbraucht die restlichen CPU-Ressourcen in einer Endlosschleife, dann zeigen sich ab diesen Werten auch Kommunikationsprobleme auf dem 1wire Bus (Modelle Raspberry „Pi B“ und „Pi B+“).

Bedarf es dennoch Abstraten von 1ms, so sollte der Einsatz eines Raspberry Pi 2 B, 3 B, 3 B+ bzw. 4 B in Erwägung gezogen werden. Die erzeugte Last wird dann vom Betriebssystem auf die 4 Cores verteilt, womit dem 1wire Treiber genügen CPU Zeit zur Bedienung des Busses verbleibt.

Grundsätzlich ist es sinnvoll, den 1wire Bus zunächst mit einem einzelnen Sensor in Betrieb zu nehmen. Auch ein Blick in das OWFS lohnt, will man den 1wire Bus zunächst ohne die Hubo C++-Library in Betrieb nehmen.

10.4 Sensoren

Grundsätzlich lassen sich über die analogen und digitalen Eingänge Hubos vielerlei Sensoren entweder direkt oder indirekt, nach einer einfachen Pegelanpassung (z.B. Widerstands-Spannungsteiler), anschließen.

Nachdem die Temperaturmessung recht häufig anzutreffen ist, seien hier die Temperatursensoren MCP9700 und MCP9701 genannt. Beide Typen können mit 5V Versorgungsspannung betrieben werden und insbesondere der MCP9701 liefert bei etwa 107°C ein Signal von 2,5V, womit er für viele Einsatzzwecke ideal an die analogen Eingänge Hubos angepasst ist.

Mit einem Messbereich der Sensoren zwischen -40°C..125°C (MCP9700) und -10°C..125°C (MCP9701) sind diese für viele Einsatzzwecke verwendbar. Aufgrund der unterschiedlichen Formfaktoren, in denen die Sensoren verfügbar sind, lassen sich auch sehr kleine Messstellen bewerkstelligen. Konsultieren Sie das Datenblatt, um den geeigneten Typ für sich herauszufinden.