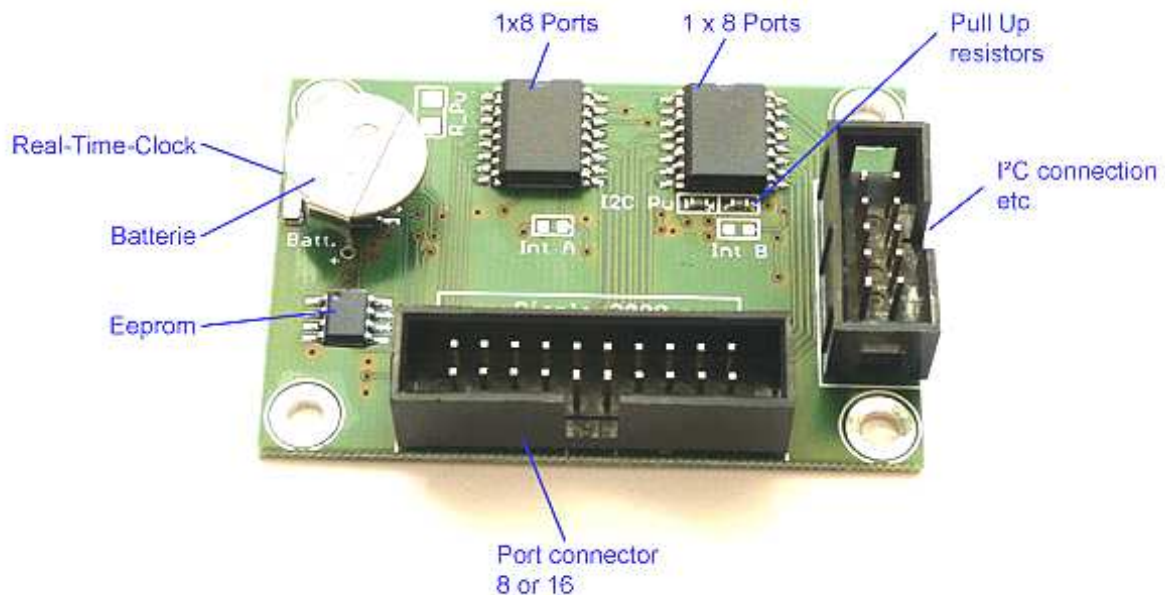


Handbuch für das I²C Modul P018

V 2.0

24. Oktober 2011



© 2011 by Peter Küsters

Dieses Dokument ist urheberrechtlich geschützt. Es ist nicht gestattet, dieses Dokument zu verändern und komplett oder Teile daraus ohne schriftliche Genehmigung von uns weiterzugeben, es zu veröffentlichen; es als Download zur Verfügung zu stellen oder den Inhalt anderweitig anderen Personen zur Verfügung zu stellen. Zuwiderhandlungen werden verfolgt.

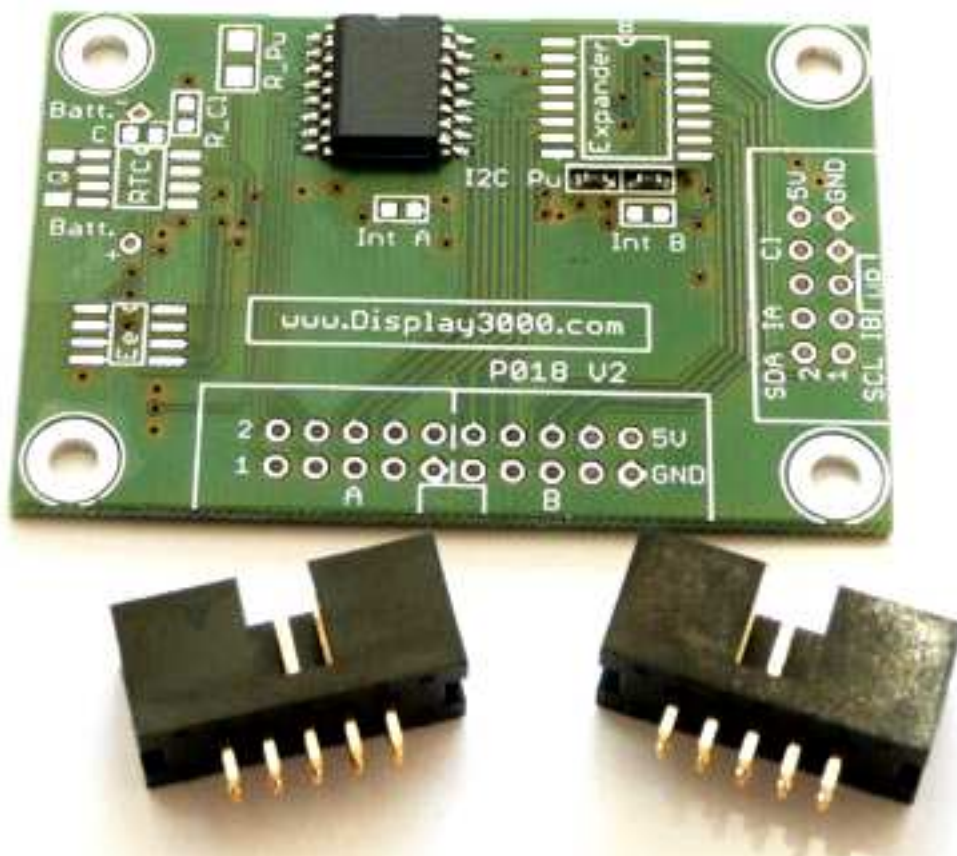
Herzlichen Glückwunsch zum Erwerb des P018-Moduls zum Anschluss an den Two Wire Anschluss (I²C®) Ihres Mikrocontrollers.

Ein Two-Wire Anschluss benötigt, wie der Name bereits impliziert, 2 Leitungen zuzüglich Masse und Vcc. Dieses Modul erweitert über eine solche Two-Wire-Verbindung die Erweiterung Ihre Mikrocontrollers um 8 oder 16 Ein-/Ausgänge. Welche und wie viele Kanäle Sie als Eingang und welche Sie als Ausgang nutzen, bleibt Ihnen überlassen.

Zusätzlich erlaubt dieses Modul noch die Erweiterung eines Mikrocontrollersystems durch eine gepufferte Echtzeituhr. Dank der Pufferung durch eine Lithiumbatterie läuft diese Uhr ständig mit (mehrere Jahre mit einer Batterie) und erlaubt es dem Mikrocontroller (z.B. beim Systemstart) sich die korrekte Uhrzeit und das korrekte Datum vom Uhrenchip zu holen.

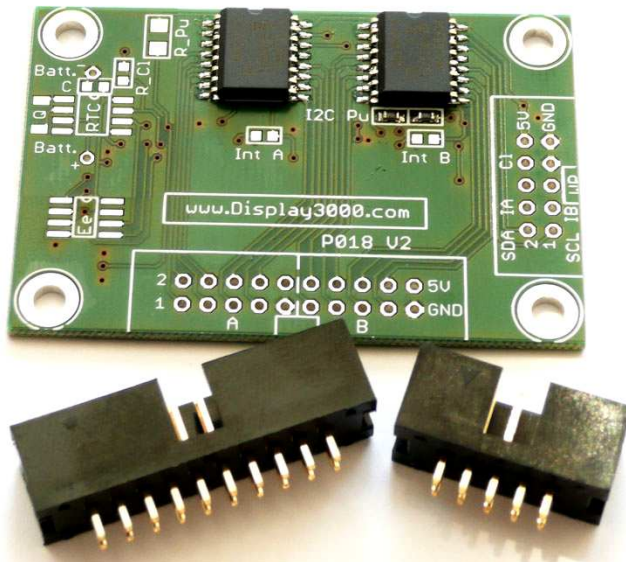
Sie erhalten die Platine als Teilbausatz. Die SMD Chips sind bereits aufgelötet, die evtl. notwendigen Wannenstecker etc. löten Sie bitte selber auf.

Abhängig vom bestellten Zubehör (Buchsen etc.) sollten Sie folgende Bauteile in Ihrer Lieferung finden:

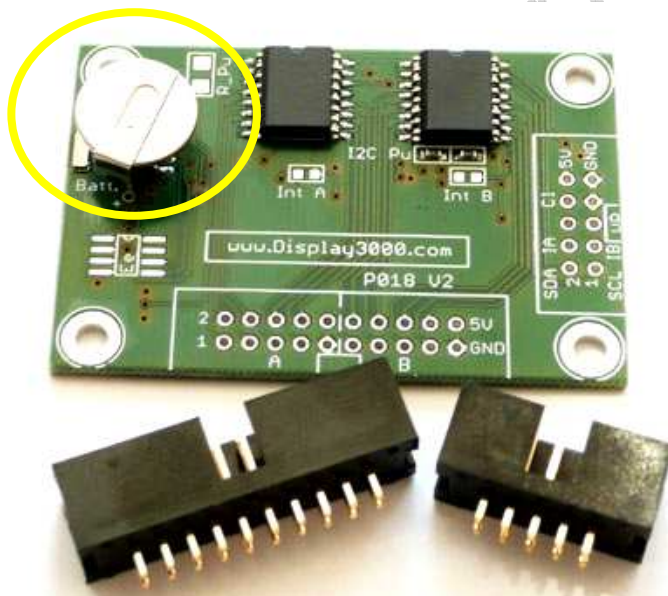


Das Foto zeigt die Platine für 8 Ein-/Ausgangskanäle. Der SMD Chip ist bereits eingelötet. Sie erhalten auch einen Streifen mit SMD Widerständen für die Portkonfiguration (mehr dazu weiter unten).

Wenn Sie das Modul mit 16 Kanälen bestellt haben, so ist der Chip doppelt vorhanden, statt des 10er Wannensteckers für die Ports, erhalten Sie einen 20er Stecker (2x10) statt eines 10er (2x5) Steckers:



Die Echtzeituhr auf dem Foto oben links (zu sehen ist hauptsächlich die Batterie – der Chip, Quarz ,etc. liegen darunter) ist komplett bestückt, sobald Sie diese Option mitbestellen.

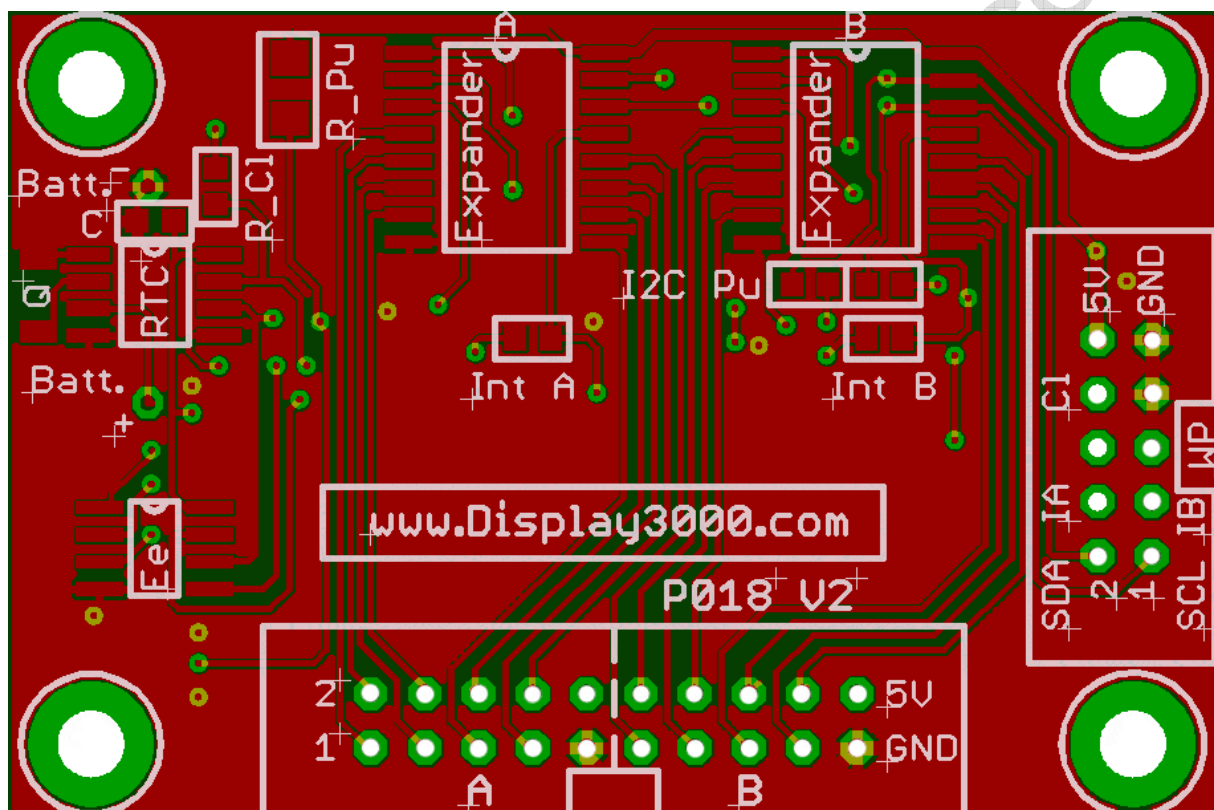


Wir teilen den I²C Expander in zwei mögliche Blöcke auf. Wenn Sie die Platine mit 8 Ports bestellt haben, dann erhalten Sie eine Platine mit einem Chip auf Block A. Das ist die linke Hälfte. Daher wird der mitgelieferte 10-polige Wannenstecker auch nur in den linken Bereich des 20-poligen Pad-Feldes gelötet (siehe übernächste Seite).

Eine Platine mit 16 Ports ist dann mit beiden Chips bestückt und wird mit dem 20-poligen Wannenstecker geliefert.

Wenn Sie mit diesen Wannensteckern arbeiten möchten (manche Kunden löten die Kabel auch lieber direkt an die Platine), dann löten Sie diese nun ein. Die Aussparung im Kunststoffkörper zeigt – wie auch durch die Bedruckung abgebildet – zum Platinenrand.

Die Platine liegt Ihnen wie folgt vor (Bestückung je nach bestellter Optionen):



Wichtig: Das Feld „R_Pu“ (oben links) kann gebrückt oder mit einem 4,7K bis 10K Widerstand bestückt werden. Es verbindet die verschiedenen Jumper-Felder für die Adresseinstellung der einzelnen Chips mit Vcc. Ohne Widerstand oder Brücke über die beiden Felder ist kein Betrieb möglich, da keine sichere Adressierung stattfinden kann.

Die Steckerbelegung der Expander-Platine

Teil A: Verbindung zum Mikrocontroller

Der Mikrocontroller wird über den Stecker am rechten Rand verbunden. Generell sind für eine I²C-Verbindung hier nur 3 Leitungen notwendig:

1. SDA (Data)
2. SCL (Clock)
3. GND (gemeinsame Masse)

In diesem Fall kommt noch eine weitere Verbindung hinzu:

4. VCC (Versorgungsspannung)

Da unsere Platine mehr noch Optionen bietet, nutzen wir auch mehr als nur diese vier, oben genannten Leitungen. Es kommen noch hinzu:

5. Interrupt-Leitung von Block A
6. Interrupt-Leitung von Block B
7. Optionales Clock-Signal von der Echtzeituhr
8. WP für ein optionales Eeprom.

Der Wannenstecker am rechten Rand ist also mit insg. 8 Leitungen belegt. Wir nutzen hier, wie auch bei unseren Modulen, den STK-500-Standard, so dass Sie diese Expanderplatine auf all unseren Mikrocontrollerplatinen mit 10er Wannenstecker (z.B. D071x, D083, D084) über ein Flachbandkabel direkt aufstecken können.

Der Stecker (und somit auch ein evtl. verwendetes Flachbandkabel) sind wie folgt belegt:

Pin	SV3
1 (Ring):	SCL (Clock)
2	SDA (Data)
3	Interrupt IC A (links)
4	Interrupt IC B (rechts)
5	WP (für optionales Eeprom)
6	Nicht beschaltet
7	Masse
8	Clock-Ausgang RTC
9	Masse
10	Vcc

Teil B: Portblock A / Portblock B

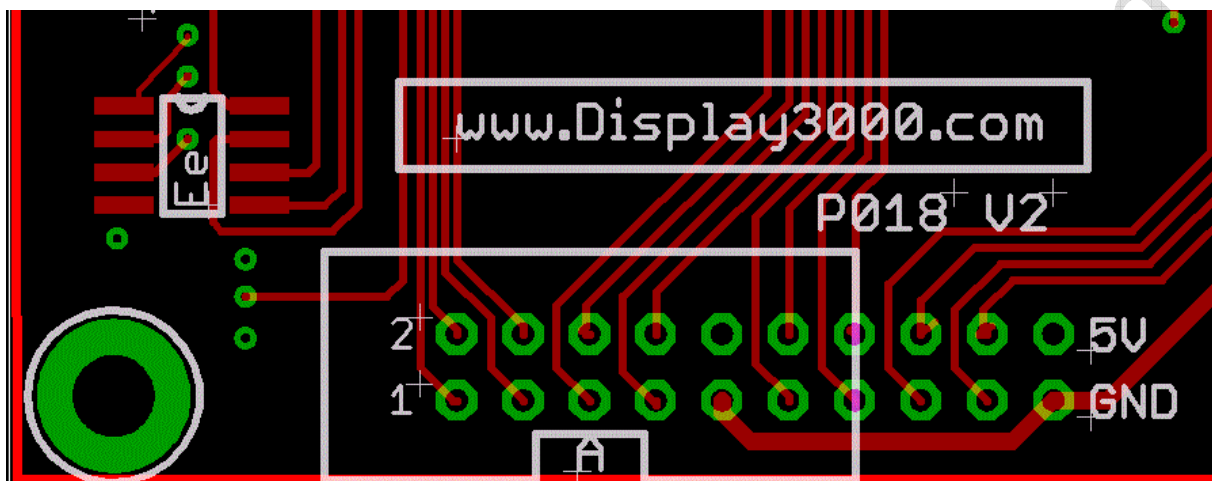
Jeder Block besteht auf 8 Ports, die, wie später noch erläutert wird, Eingang oder Ausgang sein können. Hinzu kommen noch Versorgungsspannung VCC und GND.

Bei nur einem Chip wird der 10-polige Wannenstecker links eingelötet und ist wie folgt belegt:

Pin 1 bis 8: Port 0 bis 7

Pin 9: Masse

Pin 10: VCC



Beim Vorhandensein beider Chips, ist der 20-polige Wannenstecker einzusetzen.

Die Belegung gilt dann wie folgt:

Pin 1 bis 8: Port 0 bis 7 von Chip A

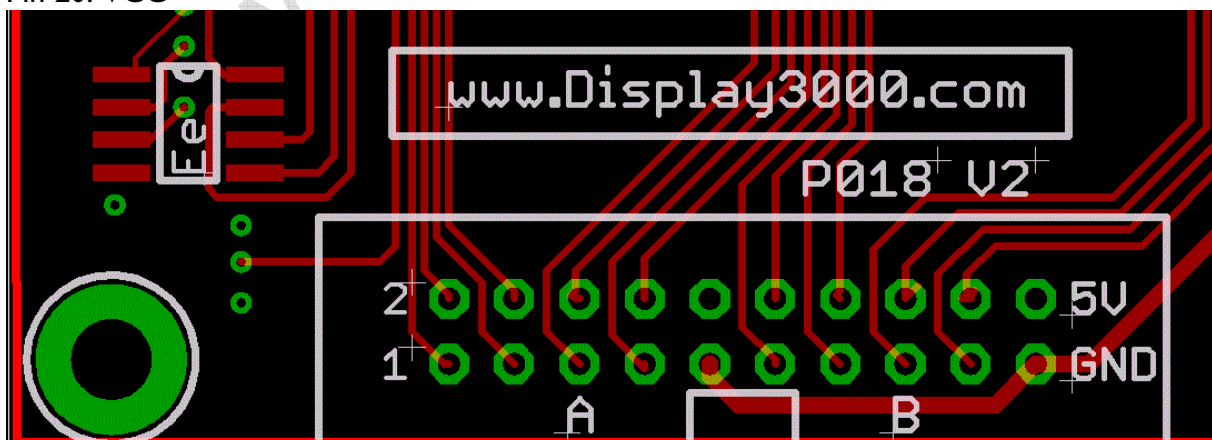
Pin 9: Masse

Pin 10: VCC

Pin 11 bis 18: Port 0 bis 7 von Chip B

Pin 19: Masse

Pin 20: VCC



Die Funktionsweise des Moduls ist schnell erläutert:

Jeder I²C-Chip (PCF 8574) bietet folgende Anschlüsse:

2 x Datenleitungen (SCL und SDA)

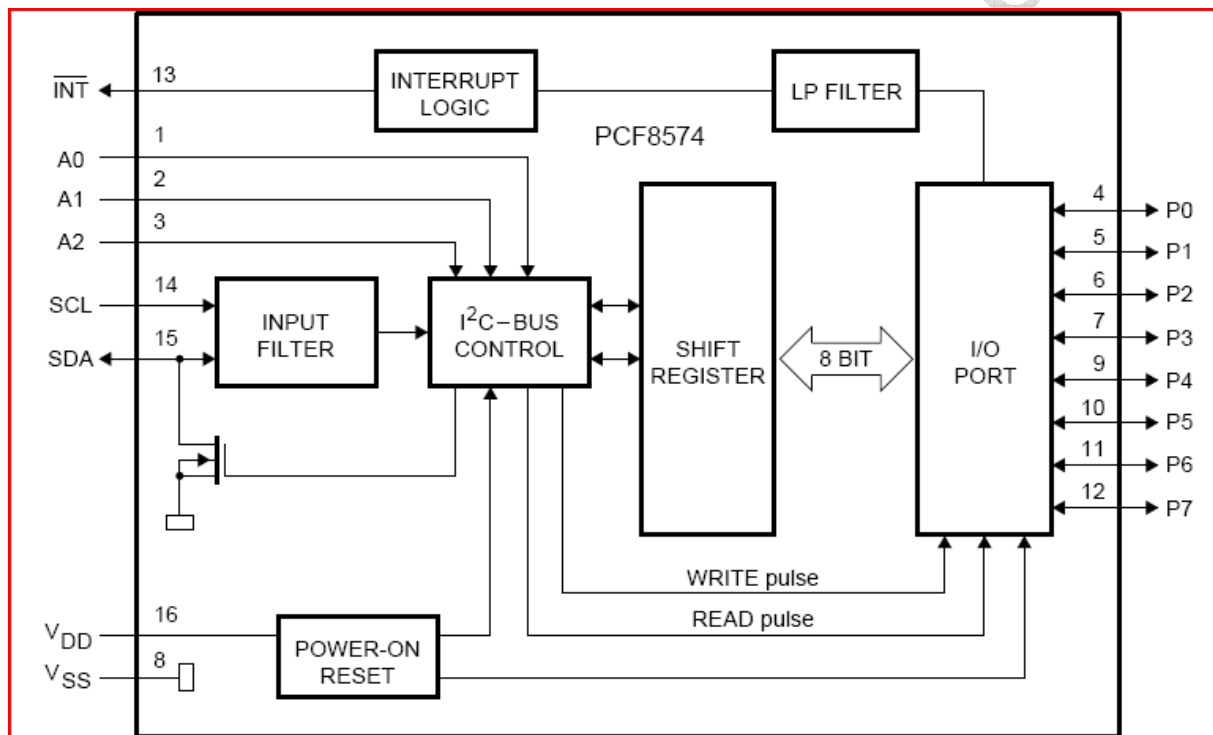
8 x Ein-/Ausgangsleitungen (P0 bis P7)

1 x Interruptleitung (/INT)

3 x Adressleitungen (A0, A1, A2)

Masse

V_{CC}



Datenleitungen:

Die beiden Datenleitungen werden über den I²C-Bus mit einem Clock-Signal (SCL) sowie einem Datensignal (SDA) versorgt.

Ein-/Ausgangsleitungen

Per Software wird von Ihnen festgelegt, welche dieser 8 Leitungen als Ein- und welche Leitungen als Ausgang fungieren sollen. Diese können dann nach Belieben gesetzt bzw. abgefragt werden. Diese 8 Ports sind quasi-bidirektionale I/O Ports (mehr dazu später), d.h. Sie können z.B. 3 Eingänge und 5 Ausgänge nutzen ... oder 7 Eingänge und ein Ausgang ... oder...

Interruptleitung

Damit der Mikrocontroller nicht permanent die Eingänge nach einer evtl. Änderung abfragen muss, bietet der Chip eine Interruptleitung an. Dies ist sehr praktisch, denn es erspart eine Menge Zeit und vereinfacht auch den Programmablauf, da man sich nur dann „kümmern“ muss, wenn es nötig ist.

Besteht zwischen dem Baustein und dem Mikrocontroller gerade kein Datenverkehr über den Bus, meldet der I²C Expander sich ändernde Eingangssignale über eine Interruptleitung weiter. Diese Leitung wird auf log 0 gezogen, wenn an einem der Eingänge eine steigende oder fallende Flanke auftritt. Der Interrupt wird dann automatisch deaktiviert, wenn das Eingangsregister entweder gelesen wurde oder wenn das Eingangssignal wieder seine ursprüngliche Lage annimmt. Letzterer Fall bedeutet aber auch, dass sich der Mikrocontroller sich nach Erhalt des Interrupts auch fix um das Auslesen der I²C Ports kümmern sollte, denn ansonsten ist das Eingangssignal, welches den Interrupt ja ausgelöst hat, evtl. nicht mehr vorhanden (z.B. ein Tasterdruck).

Diese Interruptleitung benötigt einen Pullup-Widerstand, dieser kann entweder auf unserer Platine gelötet werden, oder man schaltet (üblicherweise) einfach am entsprechenden Abfrageport am Mikrocontroller (Interrupt-Eingang) den internen Pullup-Widerstand des Mikrocontrollers ein. Die beiden Felder sind auf der Oberseite mit „Int A“ bzw. „Int B“ gekennzeichnet und normalerweise nicht bestückt.

Adressleitungen des PCF8574 / PCF8574A

Da an einem I²C Bus viele Busteilnehmer gleichzeitig verbunden sein können, muss es eine Möglichkeit geben, den „richtigen“ Teilnehmer zu adressieren, damit nicht alle Teilnehmer gleichzeitig auf den Befehl reagieren.

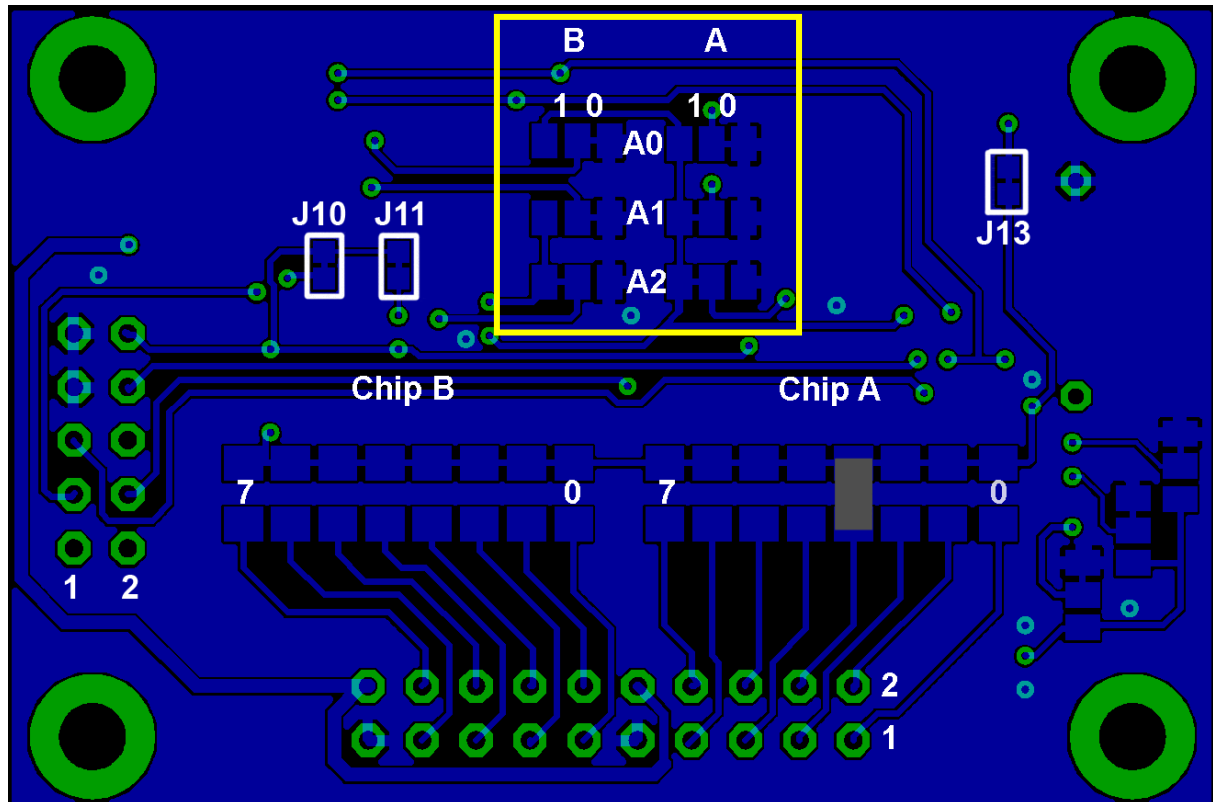
Die I²C Bausteine auf unserer Platine haben als I²C Busteilnehmer eine bestimmte Adresse. Sie sind mit 8 Bit adressierbar, wobei 5 Bits bereits fest vorgegeben sind und nicht geändert werden können. 3 Bits sind vom Anwender definierbar und von uns als Lötbrücken auf der Unterseite der Platine herausgeführt, so dass Sie dort die Adressierung bei Bedarf ändern können. Dies bietet sich dann an, wenn Sie z.B. mehrere dieser Platinen gleichzeitig nutzen möchten – dann müssen die einzelnen Chips eine unterschiedliche Adresse besitzen.

Es gibt je nach Verfügbarkeit zwei mögliche Chips zur Auswahl: den PCF8574 und den PCF8574 A. Bei unterscheiden sich in der Adressierung um zwei voreingestellte und nicht änderbare Bits. Bitte prüfen Sie, welchen der Chips Sie auf Ihrer Platine verbaut haben.

Die Adressierung lautet:

PCF8574	PCF8574A
0 0 0 A2 A1 A0 RW	0 1 1 A2 A1 A0 RW

A2 bis A0 sind durch Sie einstellbar. Mit dem niedrigsten Bit (RW) legen Sie fest, ob Sie einen Port Schreiben (RW=0) oder Lesen (RW=1) möchten. Auf dem nachfolgenden Bild sehen Sie die Rückseite der Platine. Der gelb umrandete Bereich zeigt die 2 x 3 Jumper für die mögliche Änderung der Adresseinstellung.



Die Default-Voreinstellungen der Jumper sind:

PCF8574

Linker Chip: 0 | 0 0 | 1 1 0 RW = 76 dez. (4Ch) bei Write; 77 (4Dh) bei Read

Rechter Chip: 0 | 0 0 | 1 0 0 RW = 72 dez. (48h) bei Write; 73 (49h) bei Read

bzw.

PCF8574A

Linker Chip: 0 | 1 1 1 | 1 0 RW = 124 dez. (7Ch) bei Write; 125 (7Dh) bei Read

Rechter Chip: 0 | 1 1 1 | 1 0 0 RW = 120 dez. (78h) bei Write; 121 (79h) bei Read

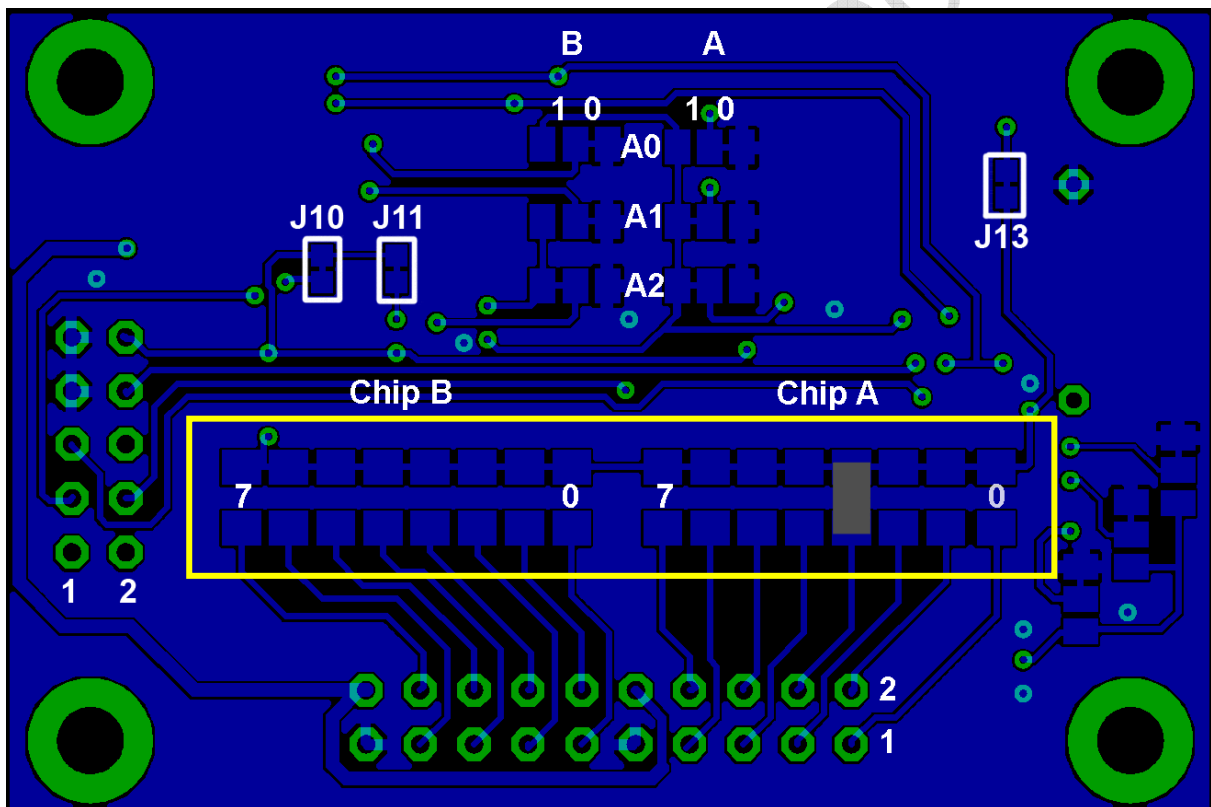
Mehr Details finden Sie im Datenblatt des Chips, welches Sie sich z.B. hier

http://www.nxp.com/documents/data_sheet/PCA8574_PCA8574A.pdf downloaden können.

Die Input / Output Ports

Ein Port kann sowohl als Eingang als auch als Ausgang genutzt werden. Durch die notwendige serielle Ansteuerung per I²C mit Adressierung etc. sind die Zugriffszeiten jedoch höher als bei direkt am Mikrocontroller genutzten Ports. Rechnen Sie mir ca. 200µS pro Zugriff auf einen dieser externen Ports.

Auf der Platinenrückseite entdecken Sie über dem Portstecker einen Bereich mit 8 leeren Feldern. Diese Felder sind vorbereitete Platzierungen für die mitgelieferten SMD Widerstände. Sie können nicht vorab durch uns eingelötet werden, da nur Sie festlegen können, welcher Port einen Pullup-Widerstand benötigt und welcher nicht. Wir haben die Felder in Bauform 1206 ausgeführt, so dass Sie nicht mit allzukleinen SMD Widerständen hantieren müssen. Selbstverständlich können Sie evtl. notwendige Pullup-Widerstände auch in bedrahteter Form außerhalb der Platine über das Anschlusskabel verbauen.



Mögliche Port-Optionen

1) OUTPUT - Der Port fungiert als Stromquelle:

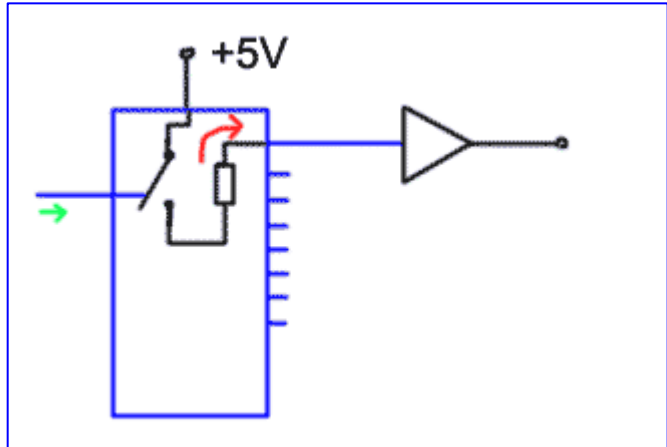
In diesem Fall fließt der Strom vom Port über z.B. einen Verbraucher nach GND.

Der Port muss dazu HI sein. Da der Port jedoch max. $100\mu\text{A}$ liefert, kann hier i.d.R. nur eine Verstärkerschaltung (z.B. Transistor) angeschlossen werden um Verbraucher zu schalten (LEDs Relais, etc.).

Der Port liefert nicht mehr als diese $100\mu\text{A}$, daher ist ein Schutz vor Kurzschlüssen hier nicht notwendig.

Schalten als High oder Low ergibt am Ausgang 5V (max $100\mu\text{A}$) oder 0V.

Dies ist die nach dem Einschalten vorhandene Default-Einstellung aller Ports.

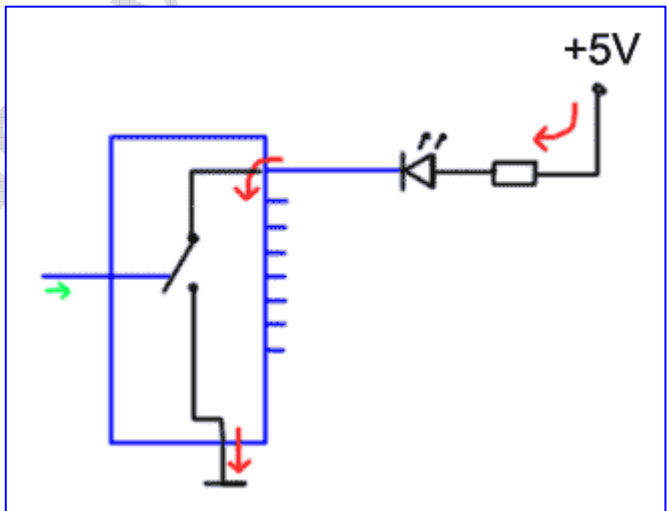


2) OUTPUT - Port als Stromsenke:

Hier fließt der Strom von Vcc (5V) über den Verbraucher (z.B. LED) über den I²C Port-Expander nach Masse.

Hierzu wird der Port von Ihnen auf Low geschaltet. Der max. zulässige Sinkstrom pro Port ist ca. 25mA. Für eine LED inkl. Vorwiderstand reicht dies aus.

Schalten als Low bedeutet: Port gegen Masse schalten (eine LED würde leuchten), High bedeutet: Kein Schalten des ports gegen Masse: LED leuchtet nicht.



3) Port als Eingang:

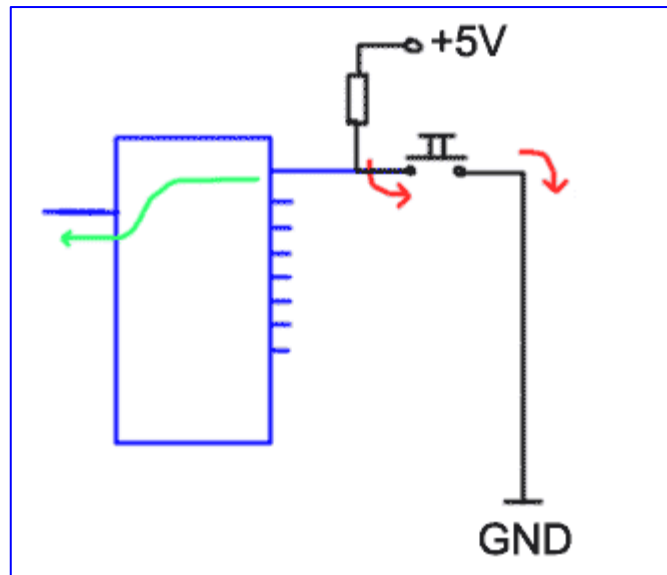
Ein PCF 8574 Port ist richtig betrachtet eigentlich immer ein Ausgang. Man kann jedoch den Ausgangs-Zustand von jedem Port lesen.

Wenn ein Port auf Low (Stromsenke) geschaltet ist, so wird dieser auch immer als Low (0) gelesen werden, denn jede evtl. dort angeschlossene Spannung wird sofort gegen Masse gezogen.

Daraus resultiert, dass ein Port, der als Eingang fungieren soll, immer erst einmal als Ausgang auf High gesetzt werden muss (s.o.: Stromquelle).

Wenn Sie nun dort z.B. Taster anschließen, die gegen Masse schließen, so können Sie diese Taster durch Lesen des Ports abfragen: High = Taster nicht gedrückt; Low = Taster gedrückt.

Soll der Port Spannungen von anderen Systemen auswerten, so bietet es sich an, an diesen Port mit einem 5-10K Pulldown-Widerstand gegen Masse anzuschließen. Somit wird dieser als Ausgang (Stromquelle) geschaltete Port (der ja nur 100 μ A liefern kann) immer als Low gelesen, da der Pulldown-Widerstand die Portspannung auf unter 3,5 Volt zieht. Ein externes Signal kann nun sicher als vorhanden (High) oder nicht vorhanden (Low) gelesen werden.



Die Echtzeituhr

Die P018 Platine kann optional mit der Option Echtzeituhr ausgestattet werden. Die Echtzeituhr benötigt zum Betrieb ebenfalls den Anschluss an den I²C Bus sowie eine Versorgungsspannung von 3,0 bis 3,6 (der von uns verbaute Typ toleriert auch 5 bis 6 Volt).

Uhrzeit / Datum

Die Echtzeituhr kann mit üblichen 3 bis 5 Volt-Systemen betrieben werden und stellt über I²C jederzeit die aktuelle Uhrzeit sowie das aktuelle Datum bereit. Selbstverständlich erkennt sie Schaltjahre und arbeitet auch dann korrekt. Um eine hohe Genauigkeit zu erhalten, arbeitet die Echtzeituhr mit einem eigenen Uhrenquarz.

Die bereitgestellten Daten sind: Sekunden, Minuten, Stunden, Tag, Monat, Jahr, Wochentag.

56 Byte RAM

Zusätzlich zur Uhrzeit stellt die Uhr auch noch einen Speicherraum von 56 Byte zur Verfügung, die Sie beliebig nutzen können.

Diese eignen sich hervorragend, um sich häufig ändernde Werte abzulegen und diese trotzdem nach einem Stromausfall noch abrufen zu können.

Zwar gibt es in Mikrocontrollern häufig einen Eeprom Bereich, der ebenfalls für solche Zwecke genutzt werden kann, jedoch hat ein Eeprom den Nachteil, dass es sich „verbraucht“: nach 100.000 Schreibzyklen ist eine Eeprom-Speicherzelle laut Datenblatt defekt. RAM kennt diese Problematik nicht und eignet sich daher hervorragend zum Ablegen sich häufig ändernder Werte.

Beispiele zur Nutzung für dieses RAMs:

- Permanentes Ablegen von Programmstatus-Daten – Nach einem Reset oder Stromausfall weiß Ihr Programm, welche Aufgaben es zuletzt durchgeführt hat
- Nutzung als Übergabespeicher von einem weiteren Mikrocontrollermodul (d.h. 2 Controller teilen sich eine Uhr und nutzen ein gemeinsames RAM von 56 Byte)
- Ablegen von Konfigurationsdaten, die der Mikrocontroller z.B. beim Start ausliest

Pufferung:

Damit sie die einmal eingestellte Uhrzeit und das Datum nicht bei Stromausfall vergisst, wird die Platine von uns mit einer Lithium-Batterie bestückt. Diese übernimmt im Falle eines Stromausfalls die Pufferung der Echtzeituhr.

Die Lithiumbatterie hat eine geringe Selbstentladung und daher kann für mind. 10 Jahre kurzfristige Stromunterbrechungen überbrücken. Wenn das Modul nicht mit Spannung versorgt wird, so arbeitet die Uhr permanent über Batterie und die Batterie kann diesen Betrieb für lange Zeit aufrechterhalten (die von uns verbaute Miniaturbatterie mit Lötanschlüssen hat

eine Lebensdauer von 10 Jahren und kann einen Spannungsausfall von insg. 1-2 Jahren kompensieren).

Ausgabe-Clocktakt:

Sehr praktisch: Der Uhrenchip kann auf Pin 7 eine programmierbare Frequenz ausgeben. Hierbei stehen die Frequenzen 1 Hz, 4 Khz, 8 Khz, 32 Khz zur Verfügung. Diese Leitung ist ein Open Drain-Anschluss und benötigt einen Pullup-Widerstand (z.B. 100 KOhm gegen 5V) – diese Leitung wird dann vom Uhrenchip gegen GND gezogen. Zum Auswerten über den Mikrocontroller, reicht es, am entsprechenden Eingang den internen Pullup-Widerstand zu aktivieren – ein externer Widerstand ist dann nicht notwendig.

Alles weitere entnehmen Sie bitte dem beigelegten Datenblatt der Echtzeituhr.

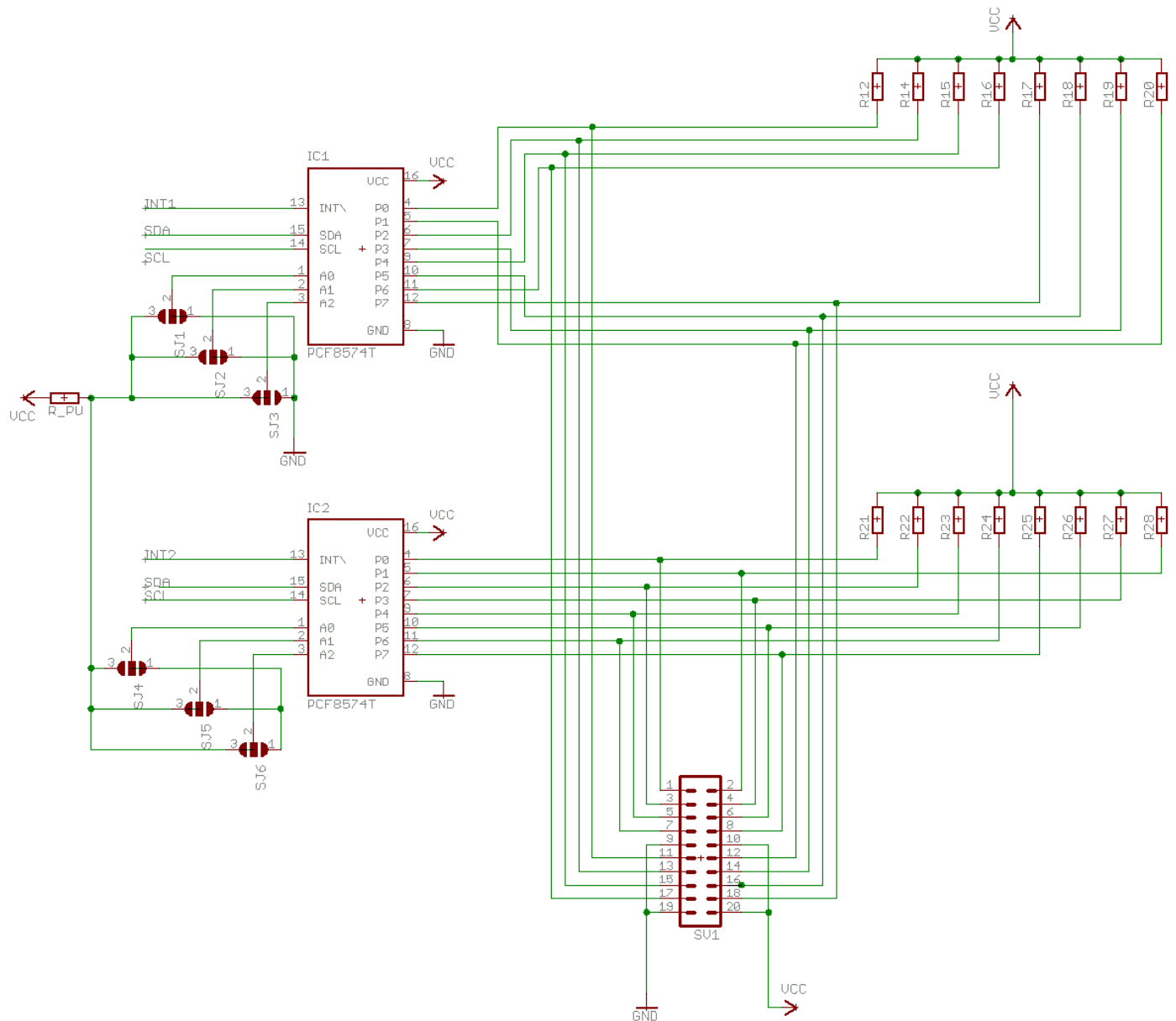
Eeprom

Die Version V2 dieser Platine bietet noch ein optionales Feld für ein Eeprom im SMD-Format S08 an. Das Feld für diesen Chip ist mit „Ee“ gekennzeichnet. Div. Eeprom Chips mit unterschiedlichen Speichergrößen sind pincompatibel und können auf diesem Feld von Ihnen als nichtflüchtiger Speicher eingesetzt werden. Ein Typ wäre z.B. ein 24AA512 , der 512Kbit Speicher anbietet (64 KByte)

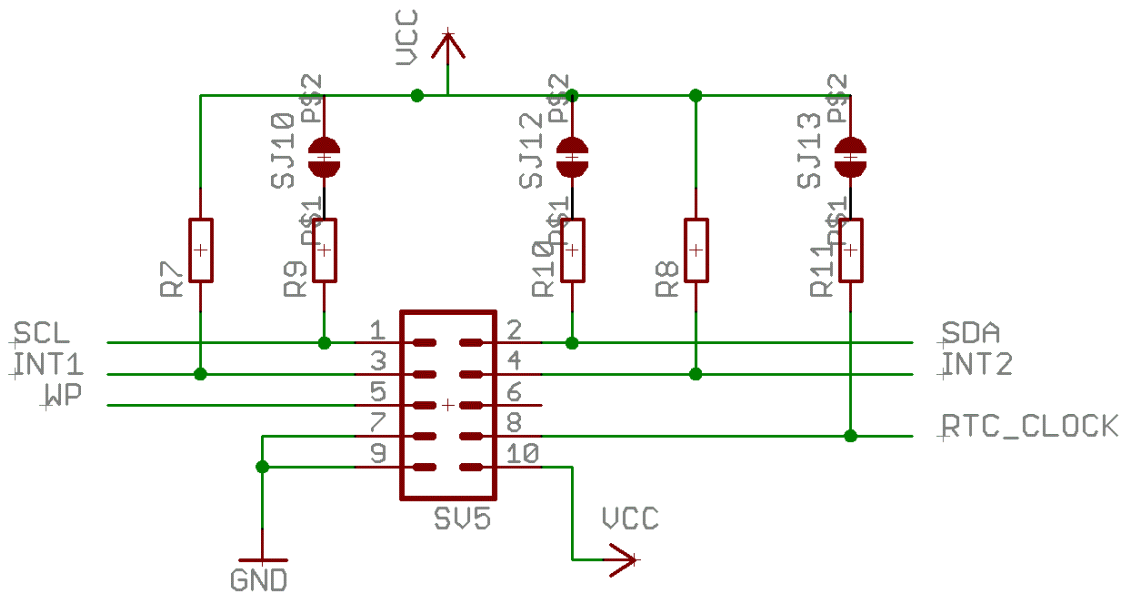
Datenblatt: <http://ww1.microchip.com/downloads/en/DeviceDoc/21203P.pdf>

Beschaltung

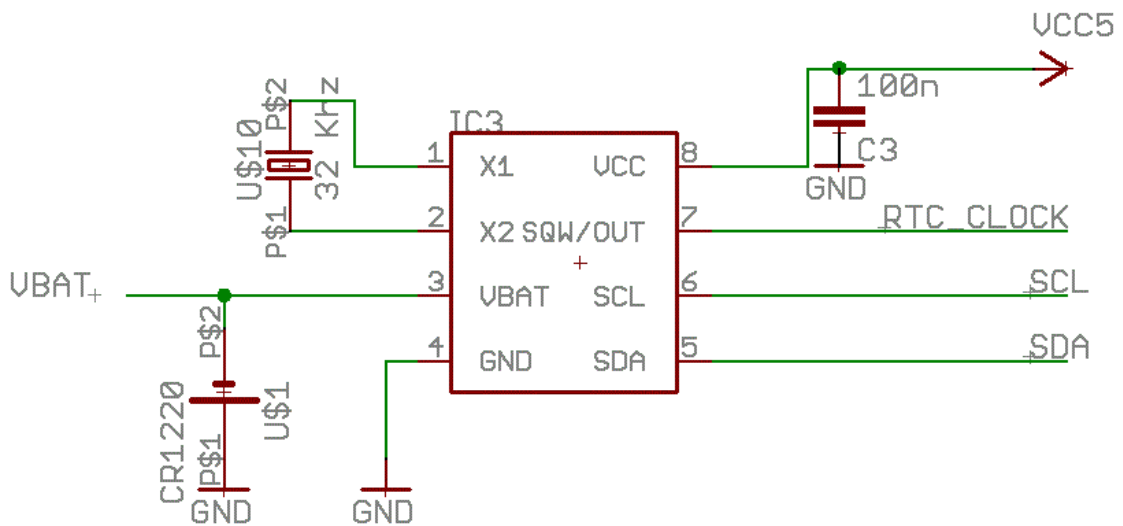
I2C Expanderchips:



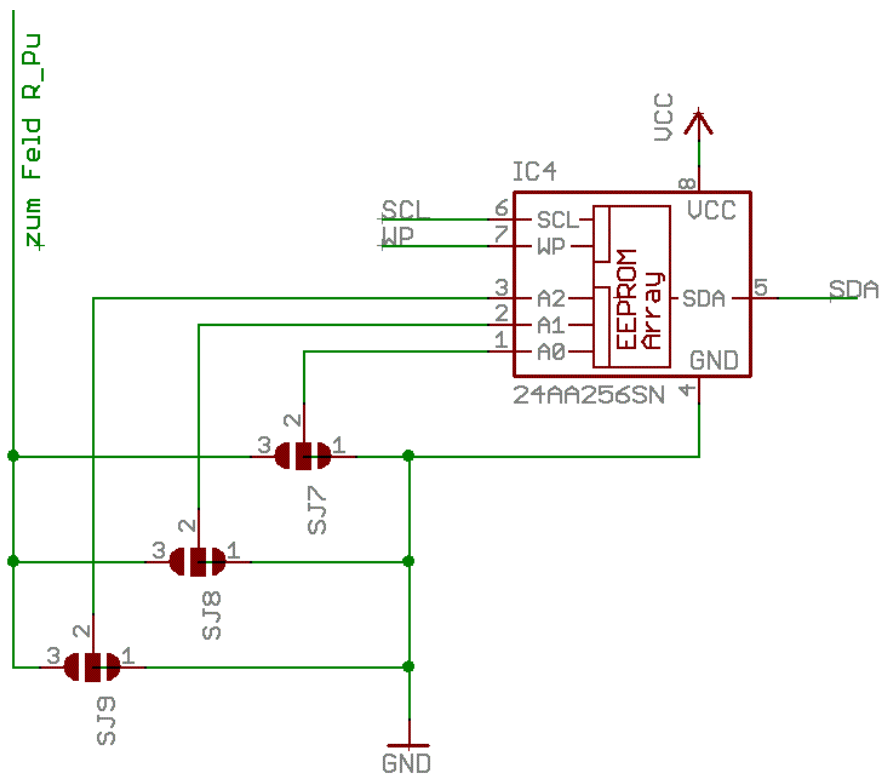
Anschlussstecker für Mikrocontroller



Echtzeituhr



Eeprom



Technische Daten

I²C Adapterplatine P018 V2

[Artikel P018](#)

Spannungsversorgung:

2,3 bis 5,5 Volt

Eingangssignale vom Mikroprozessor:

ca. 2,3 bis 5,5 Volt (*)

Stromverbrauch Standby:

ca. 10 μ A

ca. Maße:

57mm x 38mm

(*) = Die Eingänge der PCA... Chips sind auch bei niedriger Vcc tolerant bis 5,5 Volt
Dies gilt jedoch nicht für Echtzeituhr oder Eeprom.

Hersteller der Platine:

Speed IT up

Inhaber Peter Küsters

Wekeln 39

47877 Willich

Telefon: (0 21 54) 88 27 5-0

Telefax: (0 21 54) 88 27 5-22

Weitere Informationen und Updates: www.display3000.com

Autor dieses Manuals: Peter Küsters.

© aller Informationen: Peter Küsters

Haftung, EMV-Konformität

Wenn Sie diese Baugruppe durch Anlöten von Kabeln, Erweiterung bzw. Gehäuseeinbau betriebsbereit gemacht haben, gelten Sie nach DIN VDE 0869 als Hersteller und sind verpflichtet, bei der Weitergabe des Gerätes alle Begleitpapiere mitzuliefern und auch Ihren Namen und Ihre Anschrift anzugeben.

Geräte, die aus Bausätzen selbst zusammengestellt werden, sind sicherheitstechnisch wie ein industrielles Produkt zu betrachten.

Derjenige, der den Bausatz zusammenbaut und in einem Gehäuse montiert, gilt als Hersteller und ist damit selbst für die Einhaltung der geltenden Sicherheits- und EMV-Vorschriften verantwortlich.

Für Schäden die durch fehlerhaften Aufbau entstanden sind, direkt oder indirekt, ist die Haftung generell ausgeschlossen.

Bei der Lieferung von Fremdprodukten als auch Software gelten über diese Bedingungen hinaus die besonderen Lizenz- oder sonstigen Bedingungen des Herstellers.