

# PCF8574(A) Expansion Module

## Beschreibung

Der PCF8574(A) ist eigentlich die Bezeichnung des Chipsatzes, so gut wie immer findet man aber unter dem Namen die folgende Bauart eines fertigen Moduls:

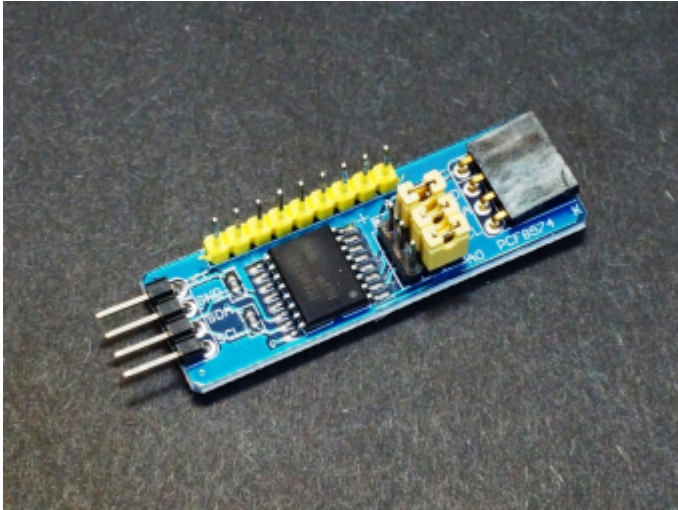


Abbildung: © <http://protosupplies.com>

Ein Chip bzw. Modul kann 8 IOs bereitstellen. Je nach Konfiguration können diese als Ein- oder Ausgang dienen. Über einen Interrupt (der an einen GPIO des Raspberry angeschlossen werden muss), werden bei Eingängen Änderungen in Echtzeit eingelesen. Es können bis zu 8 Module parallel betrieben werden (also insgesamt 64 IOs). Dazu werden diese im "Daisy Chain"-Verfahren einfach hintereinander gesteckt. Wenn das nicht ausreicht, können weitere 8 Module des Schwester-Chipsatzes PCF8574**A** verwendet werden und so bis zu 128 IOs am Raspberry bereitgestellt werden. Reicht das immer noch nicht, können zusätzlich [der Schwester-Chipsatz PCF8575](#) verwendet werden und so bis zu 256 IOs an den Raspberry angeschlossen werden!

## Datenblatt

PCF8574: <https://www.ti.com/lit/ds/symlink/pcf8574.pdf>

PCF8574A: <https://www.ti.com/lit/ds/symlink/pcf8574a.pdf>

## Hardware-Konfiguration

Das Modul wird an den I2C-Bus des Raspberry angeschlossen (PIN SDA, SCL) und benötigt zusätzlich die 3.3V Spannungsversorgung des Rasperrys (Anschluss an VCC und GND). Das Modul darf **nicht** an die 5V Spannungsversorgung des Rasperrys angeschlossen werden!

Wenn das Modul als Eingangskarte verwendet werden soll, so muss die Interrupt-Leitung (PIN INT) des Moduls noch mit einem GPIO des Rasperrys verbunden werden.

Welche Bus-Adresse das Modul verwendet, muss über 3 Jumper eingestellt werden. Möglich sind Adressen zwischen 0x20 und 0x27 (PCF8574) bzw. 0x38 und 0x3F (PCF8574A). **Achtung!** Meine Module waren an den Jumpern zwar genauso beschriftet, wie die Abbildungen auf allen Internetseiten, aber Jumper A0 und A2 waren vertauscht, also Reihenfolge genau umgekehrt!

Die Jumper werden wie folgt für die einzelnen Bus-Adressen gesetzt:

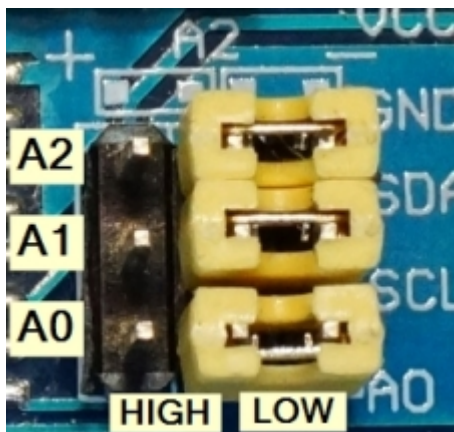


Abbildung: © <http://protosupplies.com>

Address (Hex) PCF8574	Address (Hex) PCF8574A	A2	A1	A0
0x20	0x38	LOW	LOW	LOW
0x21	0x39	LOW	LOW	HIGH
0x22	0x3A	LOW	HIGH	LOW
0x23	0x3B	LOW	HIGH	HIGH
0x24	0x3C	HIGH	LOW	LOW
0x25	0x3D	HIGH	LOW	HIGH
0x26	0x3E	HIGH	HIGH	LOW
0x27	0x3F	HIGH	HIGH	HIGH

## Software-Konfiguration

### Modul

**Add/Edit GPIO Module: mcp23017**

Name:

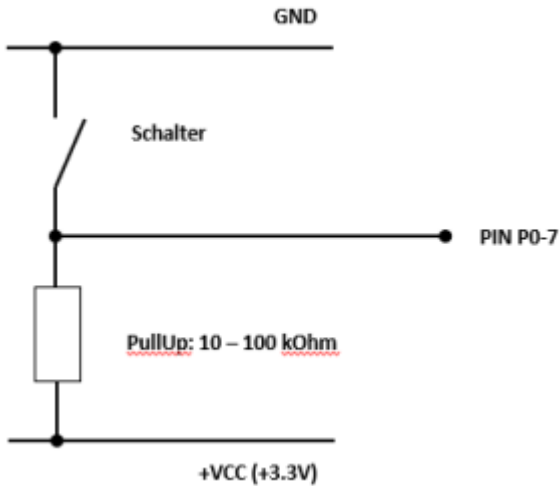
i2c Bus No.:

Chip Address:

Save  Cancel

## Eingänge

Das Modul hat eingebaute PullUp-Widerstände (keine Pulldown-Widerstände). Man verschaltet die Eingänge also so, dass sie im geschlossenen Zustand den PIN auf GND ziehen. **Ich empfehle dringend zusätzlich zu den internen Widerständen externe PullUp-Widerstände zu verwenden!** Ohne externe Widerstände konnte ich keine stabile Erkennung der Eingangszustände realisieren.



### Verschaltung mit externem PullUp-Widerstand

- Inverted: true
- Resistor: PullUp
- Interrupt: Both
- Polling Interval: 1 (wenn in Kombination mit INT-Pin)
- Polling Interval: 0.1 (ohne INT-Pin)

## Add/Edit Digital Input: pcf8574

Module Name	<input type="text" value="pcf"/>
Name	<input type="text" value="Test"/>
Pin	<input type="text" value="P1"/>
Payload for ON	<input type="text" value="ON"/>
Payload for OFF	<input type="text" value="OFF"/>
	<input checked="" type="checkbox"/> Inverted Logic Level
Enable Resistor	<input type="text" value="Pull-Up"/>
Interrupt	<input type="text" value="Both"/>
Polling Interval	<input type="text" value="1"/>
Bouncetime	<input type="text" value="180"/>
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

### Nutzung von Interrupts

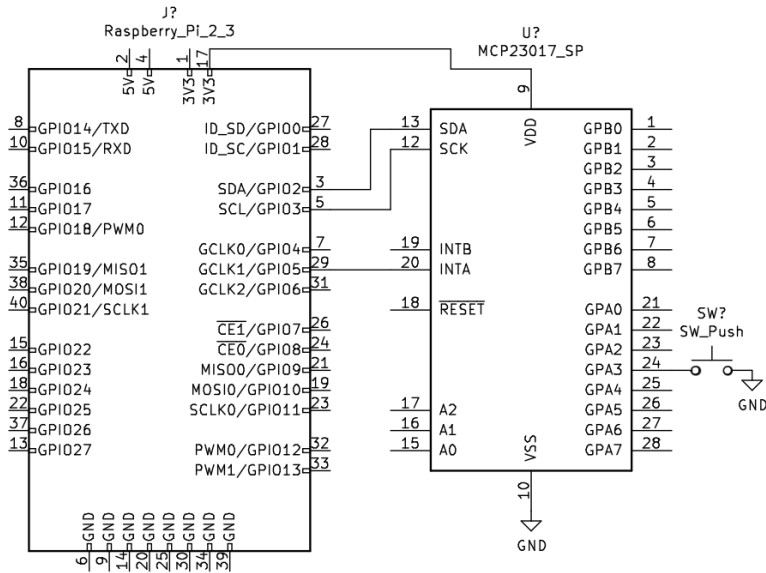
Standardmäßig werden die Zustände der Eingänge "gepollt", das heißt in einem gewissen Abstand fragt das Plugin den Zustand der Eingänge ab (ob EIN oder AUS). Das ist nicht nur sehr ineffektiv, es birgt bei sehr kurzen Eingangssignalen (z. B. ein kurzer Impuls) auch die große Gefahr, dass man ein Impuls "verpasst", wenn dieser gerade zwischen dem Poll-Intervall liegt.

Die Lösung sind sogenannte Interrupts. Interrupts signalisieren dem Betriebssystem eine Statusveränderung z. B. eines PINs. So ist es möglich den Status der Eingänge nur abzufragen, wenn sich der Status geändert hat.

Die Module haben zu diesem Zweck einen INT-Pin, der auf EIN gesetzt wird, sobald sich der Zustand eines der PINs P0-7 ändert. Der INT-Pin wird wieder auf AUS gesetzt, sobald der Zustand aller Pins P0-7 abgefragt wurde. Mit Hilfe eines GPIO-Eingangs auf dem Raspberry kann man so eine Interrupt-

gesteuerte Abfrage aufbauen. Dazu verbindet man den INT-Pin mit einem freien GPIO des Raspberry und konfiguriert diesen GPIO dann als "Interrupt für das PCF-Modul". Die GPIOs des Raspberry unterstützen echte Interrupts. Siehe unter XXXXXX.

Sobald sich nun ein Pin P0-7 des Moduls im Status ändert, wird der GPIO des Raspberry getriggert und dieser löst die Statusabfrage der einzelnen Pins P0-7 aus.



Verschaltung am Beispiel des MCP23017

### Ausgänge

- Initial: high
- Inverted: true

### Add/Edit Digital Output: pcf8574

Module Name: Relaisboard1

Name: Relais1

Pin: P0

Payload for ON: ON

Payload for OFF: OFF

Inverted Logic Level

Initial State on Startup: HIGH

Timed (in ms):

Save Cancel

From:  
<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:  
[https://wiki.loxberry.de/plugins/multi\\_io/gpio\\_module/pcf8574a\\_expansion\\_module?rev=1687527572](https://wiki.loxberry.de/plugins/multi_io/gpio_module/pcf8574a_expansion_module?rev=1687527572)

Last update: **2023/06/23 15:39**