

# MCP23017 Expansion Module

## Beschreibung

Der MCP23017 ist die Bezeichnung des Chipsatzes, so gut wie immer findet man aber unter dem Namen auch passend vorkonfektionierte Module, z. B. dieses hier von Adafruit:

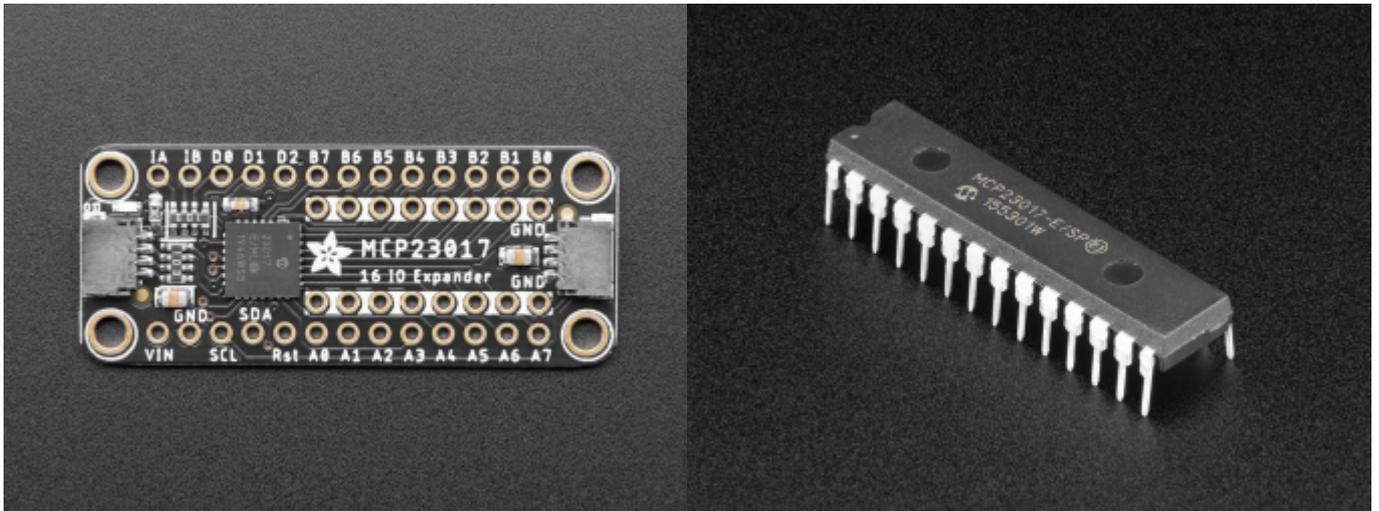


Abbildung: © <https://learn.adafruit.com/adafruit-mcp23017-i2c-gpio-expander/overview>

Ein Chip bzw. Modul kann 16 IOs bereitstellen. Je nach Konfiguration können diese als Ein- oder Ausgang dienen. Über 2 Interrupts (die jeweils an einen GPIO des Raspberry angeschlossen werden müssen), werden bei Eingängen Änderungen in Echtzeit eingelesen. Es können bis zu 8 Module parallel betrieben werden (also insgesamt 128 IOs). Dazu werden diese im "Daisy Chain"-Verfahren einfach hintereinander geschaltet.

## Datenblatt

MCP23017: <http://ww1.microchip.com/downloads/en/DeviceDoc/20001952C.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 INTA bzw. PIN INTB) des Moduls noch mit einem GPIO des Rasperrys verbunden werden.

Welche Bus-Adresse das Modul verwendet, muss über 3 PINs auf dem Modul eingestellt werden. Möglich sind Adressen zwischen 0x20 und 0x27

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

Address (Hex) MCP23017	D2/A2	D1/A1	D0/A0
0x20	LOW	LOW	LOW
0x21	LOW	LOW	HIGH
0x22	LOW	HIGH	LOW
0x23	LOW	HIGH	HIGH
0x24	HIGH	LOW	LOW
0x25	HIGH	LOW	HIGH
0x26	HIGH	HIGH	LOW
0x27	HIGH	HIGH	HIGH

## Software-Konfiguration

### Modul

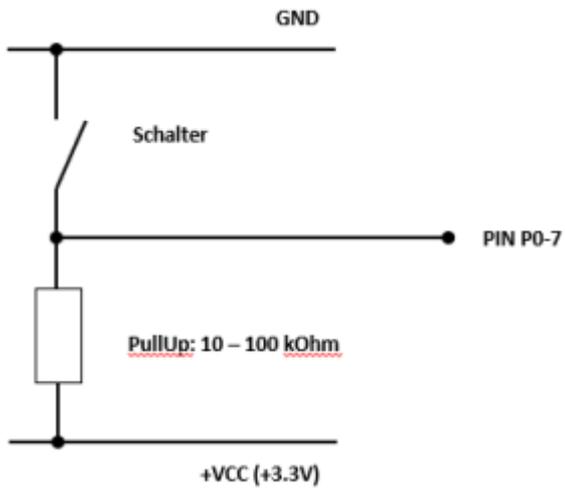


The screenshot shows a configuration window titled "Add/Edit GPIO Module: mcp23017". It contains the following fields and controls:

- Name:** A text input field with the placeholder text "Enter an unique name here".
- i2c Bus No.:** A dropdown menu currently showing "Bus 1".
- Chip Address:** A dropdown menu currently showing "0x20".
- Buttons:** Three buttons at the bottom: "Save" (with a checkmark icon), "Cancel" (with an 'X' icon), and "Scan Bus" (with a magnifying glass icon).

### 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

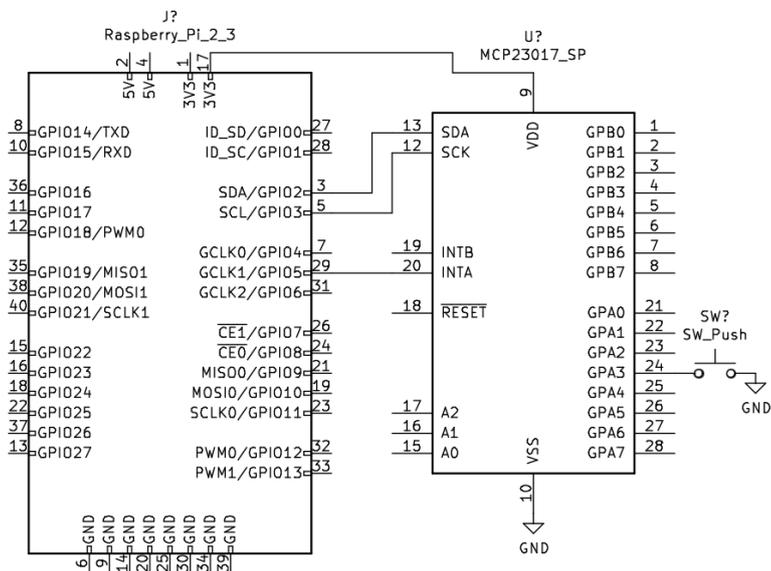
Standardmäßig werden die Zustände der Eingänge "gepollt", das heißt in einem gewissen Abstand fragt das Plugin den Zustand der Eingänge ab (ob EIN oder AUS). Das ist nicht nur sehr ineffektiv, es birgt bei sehr kurzen Eingangssignalen (z. B. ein kurzer Impuls) auch die große Gefahr, dass man ein Impuls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

Bitte ergänzen

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

Permanent link: [https://wiki.loxberry.de/plugins/multi\\_io/gpio\\_module/mcp23017\\_expansion\\_module?rev=1687527752](https://wiki.loxberry.de/plugins/multi_io/gpio_module/mcp23017_expansion_module?rev=1687527752)

Last update: **2023/06/23 15:42**