

MIKROKONTROLLER & I²C BUS

by AS



www.makerconnect.de

<https://www.makerconnect.de/resource>

Raspberry Pi Pico
I²C Bus



Copyright

Sofern nicht anders angegeben, stehen die Inhalte dieser Dokumentation unter einer „Creative Commons - Namensnennung-NichtKommerziell-Weitergabe unter gleichen Bedingungen 3.0 DE Lizenz“



Sicherheitshinweise

Lesen Sie diese Gebrauchsanleitung, bevor Sie diesen Bausatz in Betrieb nehmen und bewahren Sie diese an einem für alle Benutzer jederzeit zugänglichen Platz auf. Bei Schäden, die durch Nichtbeachtung dieser Bedienungsanleitung verursacht werden, erlischt die Gewährleistung / Garantie. Für Folgeschäden übernehmen wir keine Haftung! Bei allen Geräten, die zu ihrem Betrieb eine elektrische Spannung benötigen, müssen die gültigen VDE-Vorschriften beachtet werden. Besonders relevant sind für diesen Bausatz die VDE-Richtlinien VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860. Bitte beachten Sie auch nachfolgende Sicherheitshinweise:

- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige Person anschließen.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfwerkstätten ist das Betreiben dieser Baugruppe durch geschultes Personal verantwortlich zu überwachen.
- In einer Umgebung in der brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können, darf diese Baugruppe nicht betrieben werden.
- Im Falle einer Reparatur dieser Baugruppe, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen. Eine Reparatur des Gerätes darf nur von fachkundigen Personen durchgeführt werden.
- Spannungsführende Teile an dieser Baugruppe dürfen nur dann berührt werden (gilt auch für Werkzeuge, Messinstrumente o.ä.), wenn sichergestellt ist, dass die Baugruppe von der Versorgungsspannung getrennt wurde und elektrische Ladungen, die in den in der Baugruppe befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, muss ein Trenntrafo zur Spannungsversorgung verwendet werden
- Spannungsführende Kabel oder Leitungen, mit denen die Baugruppe verbunden ist, müssen immer auf Isolationsfehler oder Bruchstellen kontrolliert werden. Bei einem Fehler muss das Gerät unverzüglich ausser Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Es ist auf die genaue Einhaltung der genannten Kenndaten der Baugruppe und der in der Baugruppe verwendeten Bauteile zu achten. Gehen diese aus der beiliegenden Beschreibung nicht hervor, so ist eine fachkundige Person hinzuzuziehen

Bestimmungsgemäße Verwendung

- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht zulässig!
- Der Bausatz ist nur für den Gebrauch in trockenen und sauberen Räumen bestimmt.
- Wird dieser Bausatz nicht bestimmungsgemäß eingesetzt kann er beschädigt werden, was mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag etc. verbunden ist. Der Bausatz darf nicht geändert bzw. umgebaut werden!
- Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und /oder Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Der Autor dieses Tutorials übernimmt keine Haftung für Schäden. Die Nutzung der Hard- und Software erfolgt auf eigenes Risiko.

Raspberry Pi Pico – I²C Bus

Im Datenblatt des Herstellers werden verschiedene Pins angegeben um z.B. SPI Bus und den I²C Bus anzuschliessen.

- 2x SPI-Bus (Serial Peripheral Interface)
- 2x I²C-Bus (Inter-Integrated Circuit)

In diesem Teil werde ich mich auf den I²C Bus beschränken.

I²C Bus

Der I²C Bus liegt an verschiedenen Pins des Pico an. Im Einzelnen sind es diese:

I²C Bus 0: Pin 1, 2, 6, 7, 11, 12, 16, 17, 21, 22, 26 und 27 (SCL und SDA)

I²C Bus 1: Pin 4, 5, 9, 10, 14, 15, 19, 20, 24, 25, 31 und 32 (SCL und SDA)

Bei meiner Hardware arbeite ich vorzugsweise mit dem I²C Bus an den Pins 1 (SDA) und Pin 2 (SCL). Zur Anpassung von 3,3V auf 5V verwende ich einen Pegelwandler.

Die GPIOs des Pico sind auf einen Betrieb mit 3,3 V ausgelegt (nicht 5V !). Eine direkte Verbindung zwischen dem I²C Bus des Pico und meinem I²C Bus kann nur mit einem Pegelwandler erfolgen.

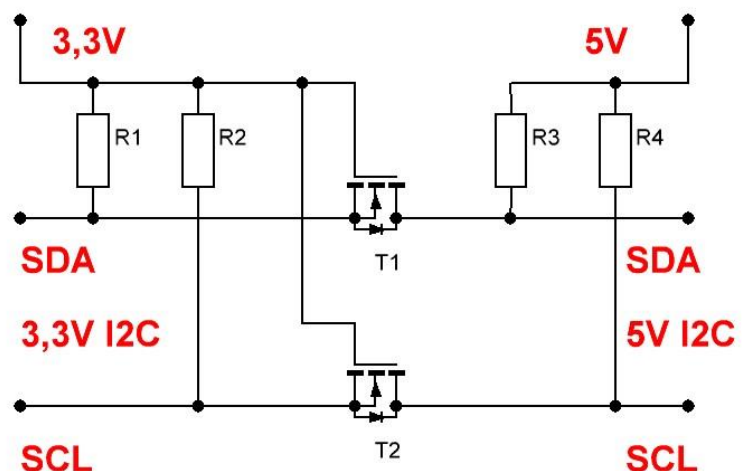
Auf meinen Platinen verwende ich meistens diese Schaltung als Pegelwandler.

Sie ist einfach aufzubauen und funktioniert super.

R1, R2, R3, R4 - 10kΩ

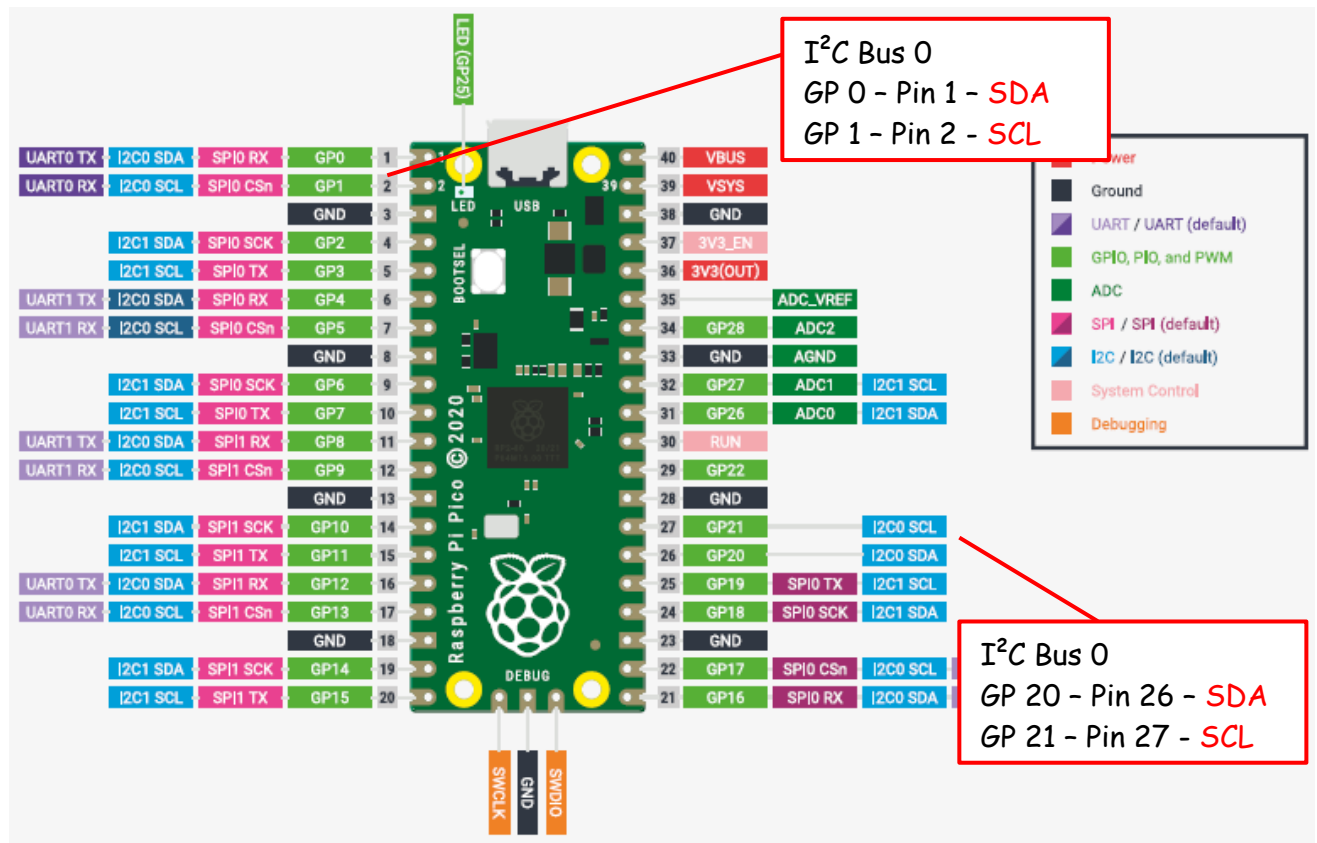
T1, T2 - 2N7000 (BSS138)

Bei den ersten Platinen habe ich meist den BSS138 verwendet. Im Netz habe ich die Info gefunden, das der 2N700 eine höhere Bus Takt zulässt. Meist arbeite ich mit 100kHz oder 400kHz. 1,2MHz habe ich bisher nicht genutzt.



Als Betriebsspannung verwende ich immer 5V. Das hat den Vorteil, nur eine Spannung für alle Module bereitzustellen und die Strombelastung der Verbindungskabel und Leiterzüge ist etwas geringer. Man sollte den Strom und den Spannungsabfall über die Verbindungen nicht unterschätzen. Werden 3,3V oder eine andere Spannung genutzt verwende ich auf der jeweiligen Platine noch mal einen extra Regler und Pegelwandler zur Anpassung.

Die Widerstände R1 - R4 haben die Aufgabe SCL und SDA auf die jeweilige Spannung zu legen. Eigentlich sind R3, R4 = 10kΩ zu gross für 5V. Teilweise habe ich auf anderen Modulen Widerstände von 4,7kΩ die ich mit einem Jumper einschalten kann. Man kann aber auch für R3 und R4 einen Wert von 4,7kΩ nehmen und dadurch auf Widerstände auf anderen Modulen verzichten.



Auszug aus dem Datenblatt des Herstellers

Gemäss Angaben im Internet kann es besser sein GP20 und GP21 zu verwenden. Dadurch werden keine anderen Ausgänge belegt. Das muss jeder für sich entscheiden.

Bei meinen Modulen erfolgt der Anschluss weiterer Module für den Raspberry Pi Pico an den Buchsen 1 - 4. Der Anschluss weiterer I²C Bus Module (5V) erfolgt an den beiden oberen Buchsen.

Die Belegung der Buchsen für den Pico und der Buchsen für den I²C Bus unterscheiden sich grundsätzlich.
 Ein vertauschen der Buchsen kann zu einer sofortigen Zerstörung des Pico führen.

Zum Beginn jedes Programmes mit dem I²C Bus verwende ich diese Zeile um die notwendigen Einstellungen zu machen. Zuerst müssen ein paar Allgemeine Angaben dazu gemacht werden:

#Variablen/Objekte

```
SCL_Pin = 1           # Angabe welcher Pin für SCL genutzt wird
SDA_Pin = 0           # Angabe welcher Pin für SDA genutzt wird
I2C_Addr_1 = 0x20     # Angabe Adresse PCF8574 mit 8xLEDs
Bus = 0               # Angabe welcher Bus verwendet wird
```

Als nächste erfolgt die Initialisierung des I²C Busses mit einer Busfrequenz von 400kHz

Initialisierung I2C, Bus 0, sda-0, scl-1

```
i2c = I2C(Bus, scl = Pin(SCL_Pin), sda = Pin(SDA_Pin), freq = 400000)
```

In diesem Beispiel verwende ich einen PCF8574 mit der Adresse 0x20.

```
PCF_Adresse_1 = pcf8574.PCF8574(i2c, I2C_Addr_1) # Beispiel Modul mit 8xLED
```

In der **Adresse PCF_Adresse_1** steckt die Info zur Bibliothek, welcher Bus verwendet wird, an welchen Pins (GPIO) der Bus angeschlossen ist und mit welcher Frequenz der Bus arbeitet.

Selbstverständlich können auch mehrere ICs angegeben werden und zusätzlich Einstellungen vorgenommen werden.

```
SCL_Pin = 1
SDA_Pin = 0
I2C_Addr_1 = 0x20 # Angabe Adresse PCF8574 mit 8xLEDs
I2C_Addr_2 = 0x21 # Angabe Adresse PCF8574 mit 4xTaster und 4x LEDs
I2C_Addr_3 = 0x27 # Angabe Adresse LCD Display mit PCF8574
Bus = 0
I2C_Rows = 4      # Angabe der Zeilen entweder 2 oder 4
I2C_Columns = 20  # Angabe der Zeichen entweder 16 oder 20
```

Initialisierung I2C, Bus 0, sda-0, scl-1

```
i2c = I2C(Bus, scl = Pin(SCL_Pin), sda = Pin(SDA_Pin), freq = 400000)
```

```
PCF_Adresse_1 = pcf8574.PCF8574(i2c, I2C_Addr_1) # Beispiel Modul mit 8xLED
```

```
PCF_Adresse_2 = pcf8574.PCF8574(i2c, I2C_Addr_2) # Beispiel Modul 4xLED 4xTaster
```

```
lcd = I2cLcd(i2c, I2C_Addr_3, I2C_Rows, I2C_Columns) # Beispiel Modul mit Display
```

In meinem Beispiel verwende ich 3 x PCF8574 mit unterschiedlichen Adressen und Nutzung.

1. PCF 8574 mit 8 x LED
2. PCF 8574 mit 4 x LED, 4 x Taster
3. PCF 8574 mit LCD

Das ganze Programm:

```
import machine
import pcf8574
from machine import I2C, Pin
from lcd_api import LcdApi
from pico_i2c_lcd import I2cLcd
import utime
```

#Variablen/Objekte

```
SCL_Pin = 1
SDA_Pin = 0
I2C_Addr_1 = 0x20      # Angabe Adresse PCF8574 mit 8xLEDs
I2C_Addr_2 = 0x21      # Angabe Adresse PCF8574 mit 4xTaster und 4x LEDs
I2C_Addr_3 = 0x27      # Angabe Adresse LCD Display mit PCF8574
Bus = 0
I2C_Rows = 4           # Angabe der Zeilen entweder 2 oder 4
I2C_Columns = 20       # Angabe der Zeichen entweder 16 oder 20
# Initialisierung I2C, Bus 0, sda-0, scl-1
i2c = I2C(Bus, scl = Pin(SCL_Pin), sda = Pin(SDA_Pin), freq = 400000)

PCF_Adresse_1 = pcf8574.PCF8574(i2c, I2C_Addr_1) # Beispiel Modul mit 8xLED
PCF_Adresse_2 = pcf8574.PCF8574(i2c, I2C_Addr_2) # Beispiel Modul 4xLED 4xTaster
lcd = I2cLcd(i2c, I2C_Addr_3, I2C_Rows, I2C_Columns) # Beispiel Modul mit Display
```

```

lcd.clear()                # LCD löschen
# Anzeige Text
lcd.move_to(0,0)           # Gehe zu Spalte/Zeile
lcd.putstr('Anzeige Tasten') # Angabe Text
lcd.move_to(0,1)           # Gehe zu Spalte/Zeile
lcd.putstr('=====')      # Angabe Text
lcd.move_to(0,2)           # Gehe zu Spalte/Zeile
lcd.putstr('Taste xx')      # Angabe Text

while True:                # Endlos Schleife Beginn
    PCF_Adresse_1.port = 0xff # alle LED aus 0xff
    PCF_Adresse_2.port = 0xff # alle LED aus 0xff
    if (PCF_Adresse_2.pin(1) == 0):
        PCF_Adresse_1.pin(4, 1) # LED an 0
        PCF_Adresse_1.pin(5, 0) # LED an 0
        lcd.move_to(6,2)         # Gehe zu Spalte/Zeile
        lcd.putstr('gedr\365ckt  ') # Angabe Text \365 Sonderzeichen für ü
    else:
        PCF_Adresse_1.pin(4, 0) # LED aus 1
        PCF_Adresse_1.pin(5, 1) # LED an 0
        lcd.move_to(6,2)         # Gehe zu Spalte/Zeile
        lcd.putstr('nicht gedr\365ckt') # Angabe Text \365 Sonderzeichen für ü

```

Funktion des Programmes:

Auf dem ersten Modul werden 8 LEDs angesteuert.

Auf dem zweiten Modul werden 4 LEDs angesteuert. Mit den 4 Tastern können verschiedene LEDs auf den beiden Modulen geschaltet werden.

Auf dem dritten Modul erfolgt die Anzeige eines Textes und ob eine Taste gedrückt wurde.

Mit der Angabe `\365` erfolgt die Anzeige des Buchstaben `ü`.

Es gibt aber noch ein Problem zu klären.

Welche Adresse hat mein IC und wie kann ich es am besten ermitteln?

Im Netz werden verschiedene Programme dazu veröffentlicht. Habe verschiedene Programme getestet und sie funktionieren fast alle super. Die Adresse wird auf dem PC angezeigt.

Mit diesem Programm können auch mehrere Geräte am Bus ermittelt werden.

```

from machine import Pin, I2C
SCL_Pin = 1          # Angabe Pin SCL (Anpassen)
SDA_Pin = 0          # Angabe Pin SDA (Anpassen)
Bus = 0              # Angabe Bus Nr. (Anpassen)
i2c = I2C(Bus, scl = Pin(SCL_Pin), sda = Pin(SDA_Pin), freq = 400000)
# Scanne Bus
print('Scanne I2C Bus...')
devices = i2c.scan()
# Scanergebnis ausgeben
if len(devices) == 0:
    print('Kein I2C-Gerät gefunden!')

```

else:

```
print('I2C-Geräte gefunden:', len(devices))  
for device in devices:  
    print('Dezimale Adresse:', device, '| Hexadezimale Adresse:', hex(device))
```

Mit diesem Programm wird die Dezimale und Hexadezimale Adresse angezeigt.

Ich hoffe dass ich euch die Anwendung des I²C Busses auf dem Pico etwas näher gebracht habe. Im Grunde sind die Programme in den einzelnen Sprachen relativ identisch, trotzdem gibt es Unterschiede im Aufbau und Nutzung der Bibliotheken.

Einige Teile des Textes wurden zur besseren Übersicht farblich gestaltet.

Die Nutzung erfolgt auf eigenes Risiko.

Ich wünsche viel Spaß beim Bauen und programmieren

Achim

myroboter@web.de

<http://www.elektronik-kompodium.de/sites/raspberry-pi/2611031.htm>

<http://www.elektronik-kompodium.de/sites/raspberry-pi/2612271.htm>