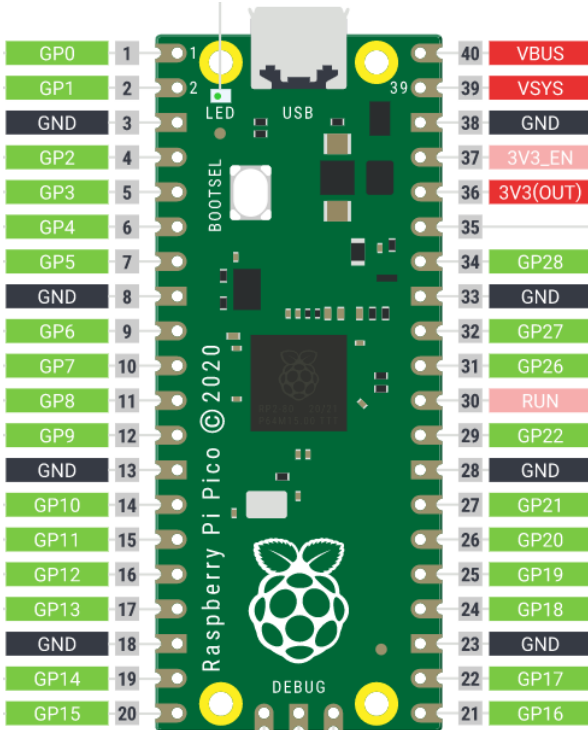


LED-Matrix - Cheat Sheet

1 Hardware

RaspberryPi Pico - Pinout



RaspberryPi Pico to PCB Mapping

Schiebeschalter: **GPIO9**

Joystick: **GPIO2,3,6,7,8**

Neopixel LEDs: **GPIO19**

LED-Matrix Index Layout

```
56 57 58 59 60 61 62 63
48 49 50 51 52 53 54 55
40 41 42 43 44 45 46 47
32 33 34 35 36 37 38 39
24 25 26 27 28 29 30 31
16 17 18 19 20 21 22 23
 8  9 10 11 12 13 14 15
 0  1  2  3  4  5  6  7
```

2 Software

2.1 MicroPython Grundlagen

Ein Programm wird von oben nach unten gelesen und auch in dieser Reihenfolge ausgeführt. Python benutzt Enrückung um Code in Blöcke zu unterteilen.

Kommentar

```
# ich bin ein Kommentar
```

Print

```
# Print
print("Hello World")
print(5)
```

Schlafen

```
from utime import sleep_ms

# Schlafen fuer 1'000 Millisekunden = 1 Sek
sleep_ms(1000)
```

Variable

```
my_number = 5
print(my_number) # prints 5

my_number = my_number + 3
print(my_number) # prints 10
```

Variablentypen

```
a = 5 # number
c = "Hello World!" # string

# boolean
d = True
e = False
g = (2 > a) # False
h = (5==a) # True

# lists
list1 = ["Hi", "you"]
list2 = [1, 5, 7, 3]
print(list1[0]) # prints first element "Hi"
print(len(list1)) # prints length "2"
```

If - Else

```
if number > 10:
    print("Groesser als 10")
elif number < 5:
    print("Kleiner als 5")
else:
    print("Zwischen 5 und 10")
```

Schleifen

```
# for Schleife
for i in range(1, 5):
    # i Werte: 1,2,3,4
    print(i)

for i in range(5):
    # i Werte: 0,1,2,3,4
    print(i)

mylist = [2,4,6,7]
for i in range(len(mylist)):
    # i Werte: 0,1,2,3 (len=Laenge=4)
    print(i)

for i in mylist:
    # i Werte: 2,4,6,7
    print(i)

# while Schleife
while number < 10:
    number += 1

# Dauerschleife
while True:
    ...
```

Funktionen

```
# Funktionsdefinition
def addieren(number1, number2):
    return number1 + number2

#Funktion aufrufen
addieren(1, 2) # Resultat ist 3

# Beispiel 2
def welcome(name):
    phrase = "Hello, " + name + "!"
    print(phrase)

welcome("Sarah") # prints "Hello, Sarah!"
```

2.2 RaspberryPi Pico

GPIO Control

```
from machine import Pin

# Definiere GPIO0 als Output (Ausgabe)
p0 = Pin(0, Pin.OUT)

# Setze den Output Wert auf 0 oder 1
p0.value(0)
p0.value(1)

# Definiere GPIO5 als Input (Eingabe)
p1 = Pin(5, Pin.IN)

# Lese den Wert von GPIO5
p1.value()
```

RTC (real time clock)

Der RaspberryPi Pico hat eine Echtzeituhr. Mit dieser kann man physikalische Zeit messen. Die Zeit muss jedoch richtig eingestellt werden, jedes mal wenn der Pico neu eingesteckt wird.

```
from machine import RTC

# RTC (real time clock) initialisieren
rtc_obj = RTC()

# datetime format:
# (Jahr, Monat, Tag, Wochentag, Stunde,
# Minute, Sekunde, Subsekunde)

# Lese die Zeit vom RTC
datetime = rtc_obj.datetime()
hour = datetime[4]
minutes = datetime[5]
seconds = datetime[6]

# Setze die Zeit vom RTC
rtc_obj.datetime((2000, 1, 1, 1, hour,
                    minutes, seconds, 0))
```

NeoPixel Library - Beispiel 1

```
from machine import Pin
from neopixel import NeoPixel

# GPIO Pin verbunden mit den LEDs
gpio = 19

# Anzahl LEDs
led_count = 64

# Initialisiere NeoPixel
np = NeoPixel(Pin(gpio, Pin.OUT), led_count)

# Setze den RGB Wert der ersten LED
np[0] = [50,0,0]

# Setze den RGB Wert der zehnten LED
np[9] = [0,25,25]

# Farbwerten an LEDs senden
np.write()
```

NeoPixel Library - Beispiel 2

```
from machine import Pin
from neopixel import NeoPixel

# Initialisiere NeoPixel
np = NeoPixel(Pin(19, Pin.OUT), 64)

# Erstelle Liste fuer LEDs, welche rot
# leuchten sollen
red_leds = [42,45,50,53]
# Erstelle Liste fuer LEDs, welche blau
# leuchten sollen
blue_leds = [10,11,12,13,17,22]

# Setze die LEDs aus den Listen im Neopixel
for i in red_leds:
    np[i] = [50,0,0]
for i in blue_leds:
    np[i] = [0,0,50]

# Farbwerte an LEDs senden
np.write()
```

2.3 ZHAW Module

Lade die Module unter folgendem Link herunter:

ledmatrix.py

```
from utime import sleep_ms
from zhaw_led_matrix import (
    ColorTable,
    LedMatrix,
    PixelColor,
)

# Kreiere eigene Farbe
my_color = PixelColor(2, 211, 13)

# LED Matrix Objekt
matrix = LedMatrix(8, 8)

# Setze Helligkeit
matrix.set_brightness(20)

# Setze Pixel x=0,y=1
# ColorTable beinhaltet 16 Farben
matrix[0, 1] = ColorTable.YELLOW

# Sende Farbwerte an LED-Matrix
matrix.apply()
sleep_ms(1000)

# Alle Neopixel auf blau setzen
matrix.fill(ColorTable.BLUE)
matrix.apply()
sleep_ms(1000)

# Alle Farben auf Null setzen
matrix.clear()
matrix.apply()
sleep_ms(1000)

# Liste von Pixel auf einen Farbwert setzen
pixel_list = [(5, 1), (5, 2), (6, 1)]
matrix.draw_list(pixel_list, ColorTable.GREEN)
matrix.apply()
sleep_ms(1000)

# Linien zeichnen:
# Gerade Linie
matrix.draw_line(
    (0, 0), (0, 4), ColorTable.ORANGE
)
# Diagonale Linie
matrix.draw_line(
    (0, 7), (5, 5), ColorTable.PINK
)
matrix.apply()
```

button.py

```
from zhaw_led_matrix import Button

# Button initialisieren
btn = Button()

# Verhalten bei einem button Ereignis festlegen
def left(_):
    print(".",)
    btn.set_left_handler(left)

# Werte vom Joystick & Schiebeschalter einlesen
btn.up.value()
btn.down.value()
btn.left.value()
btn.right.value()
btn.center.value()
btn.switch.value()
```

characters.py

```
from utime import sleep_ms
from zhaw_led_matrix import (
    CharacterTable,
    ClockTable,
    ColorTable,
    LedMatrix,
)

# LED-Matrix initialisieren
matrix = LedMatrix(8, 8)
matrix.set_brightness(20)

# Die 10 von der Minutenanzeige in blau
# leuchten lassen
matrix.draw_list(
    ClockTable.MIN_TEN, ColorTable.BLUE
)
matrix.apply()
sleep_ms(1000)

# Die 10 von der Stundenanzeige in gruen
# leuchten lassen
matrix.clear()
matrix.draw_list(
    ClockTable.HOUR_TEN, ColorTable.GREEN
)
matrix.apply()
sleep_ms(1000)

# Buchstabe A auf LED-Matrix ausgeben in rot
matrix.clear()
matrix.draw_list(
    CharacterTable.A, ColorTable.RED
)
matrix.apply()
```

LED-Matrix Layout (X,Y)

```
[0,7] [1,7] [2,7] [3,7] [4,7] [5,7] [6,7] [7,7]
[0,6] [1,6] [2,6] [3,6] [4,6] [5,6] [6,6] [7,6]
[0,5] [1,5] [2,5] [3,5] [4,5] [5,5] [6,5] [7,5]
[0,4] [1,4] [2,4] [3,4] [4,4] [5,4] [6,4] [7,4]
[0,3] [1,3] [2,3] [3,3] [4,3] [5,3] [6,3] [7,3]
[0,2] [1,2] [2,2] [3,2] [4,2] [5,2] [6,2] [7,2]
[0,1] [1,1] [2,1] [3,1] [4,1] [5,1] [6,1] [7,1]
[0,0] [1,0] [2,0] [3,0] [4,0] [5,0] [6,0] [7,0]
```

3 Links

Download Links:

- Alle Workshop Materialien (Dokumente & Programme):
<https://github.com/InES-HPMM/LED-Matrix-Workshop>
- Thonny herunterladen & installieren:
<https://thonny.org/>

Anleitungen:

- Getting startet with Raspberry Pi Pico:
<https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/0>
- Raspberry Pi Pico: WS2812 als Lauflicht programmieren:
<https://www.elektronik-kompendium.de/sites/raspberry-pi/2703111.htm>