

Workshop 9 : Physical Computing

HILF! 2024

Temperature and Humidity
Environmental Measurement & Monitoring

Universität
Rostock



Traditio et Innovatio

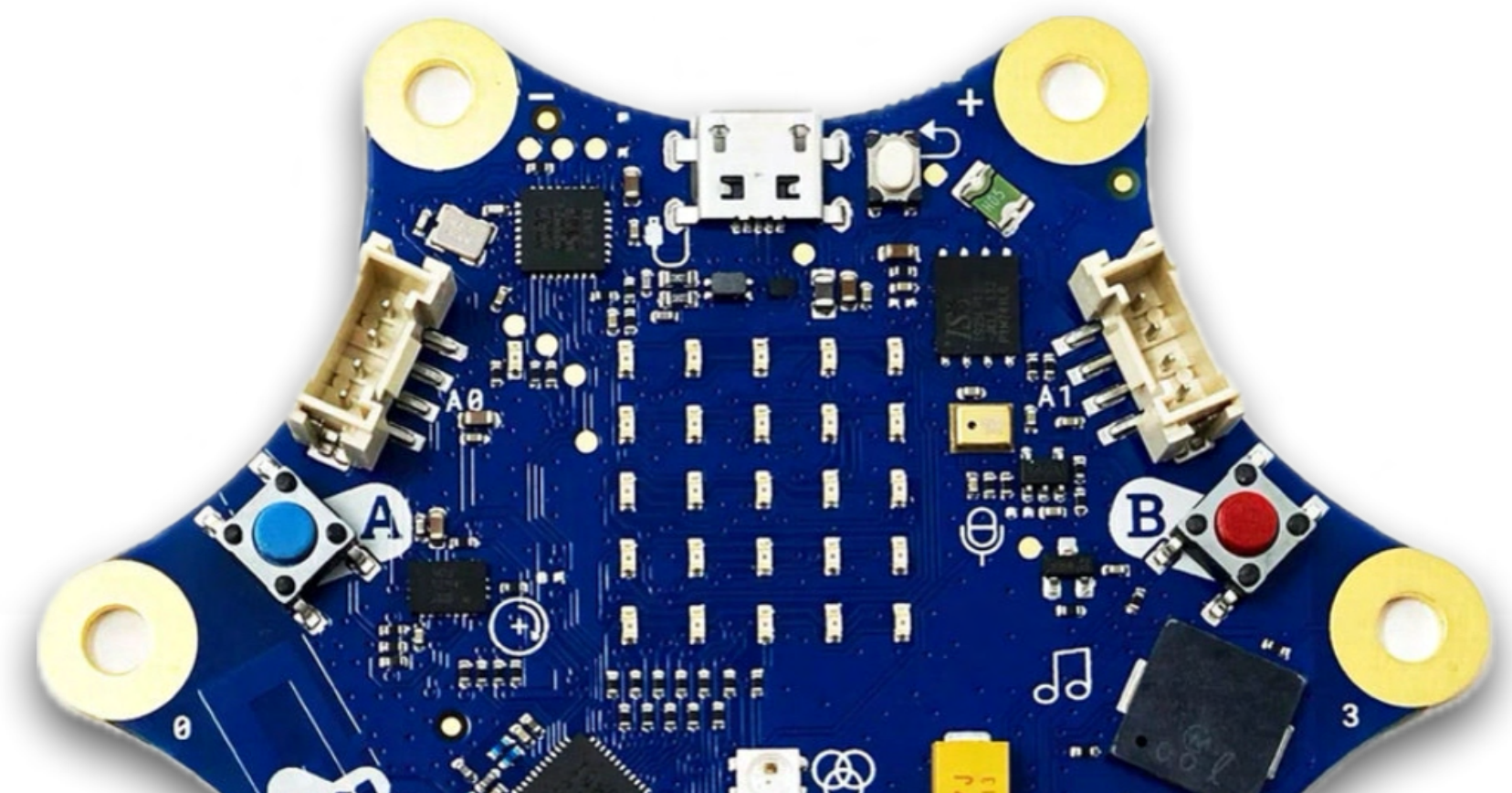
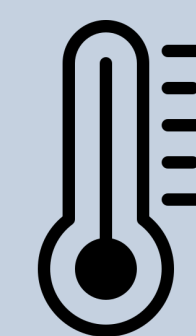
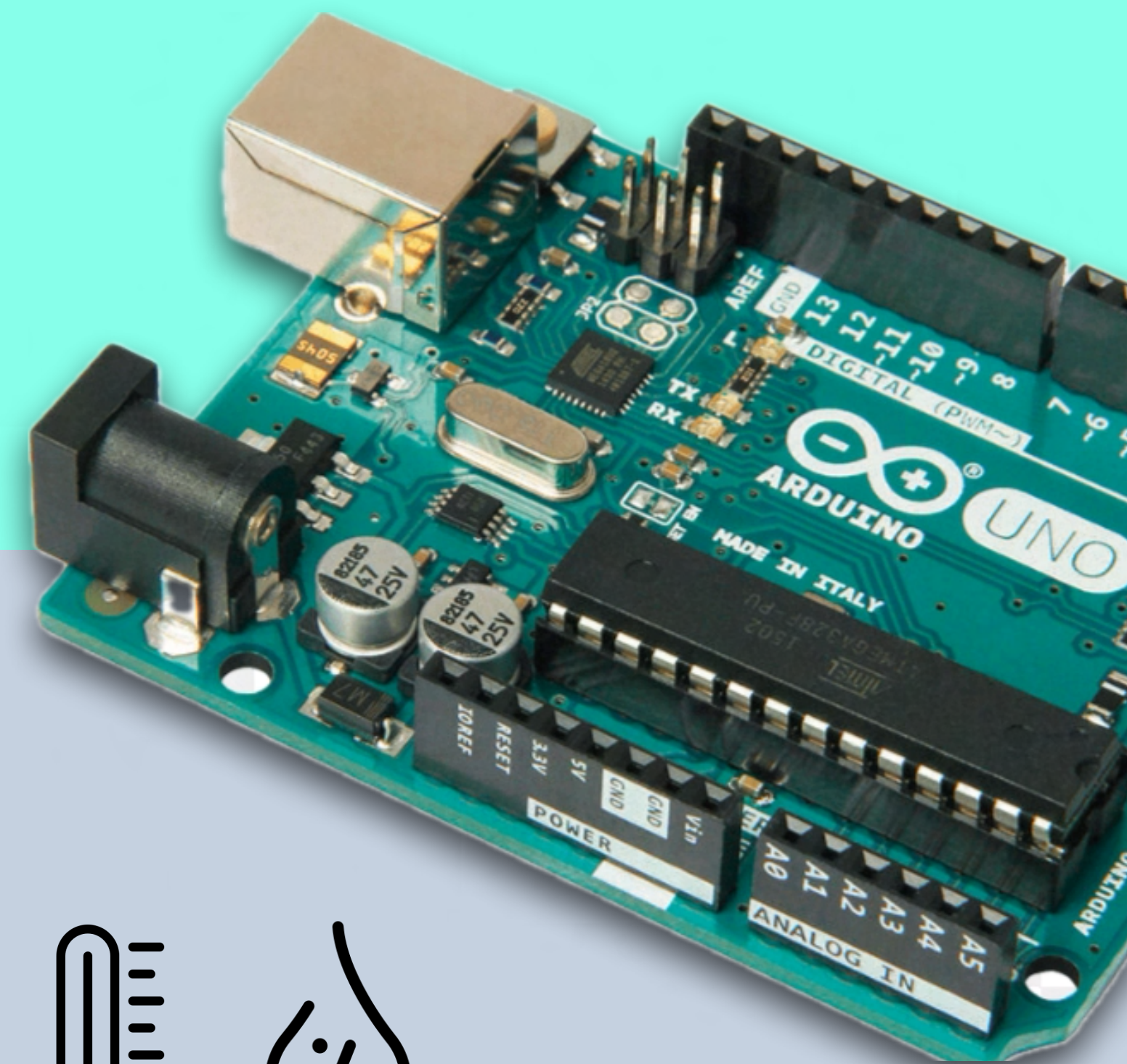


CALLIOPE

Roland Schulz, Engineer and STEM Teacher

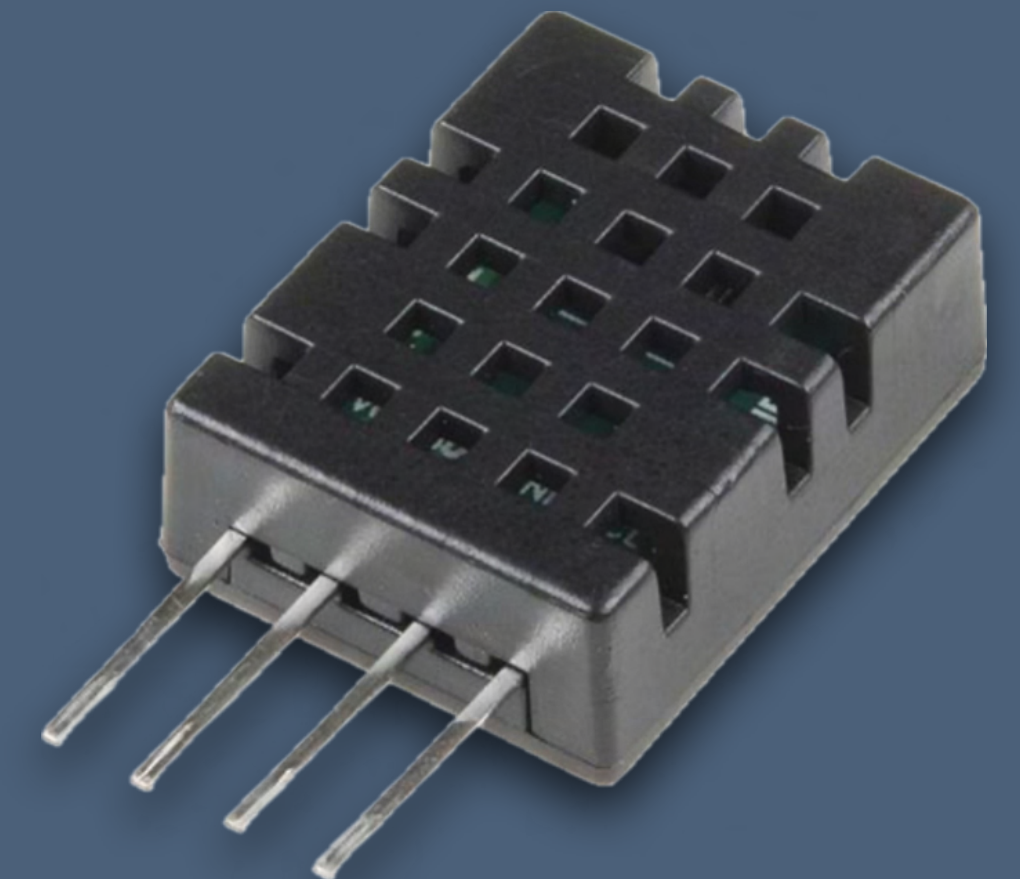
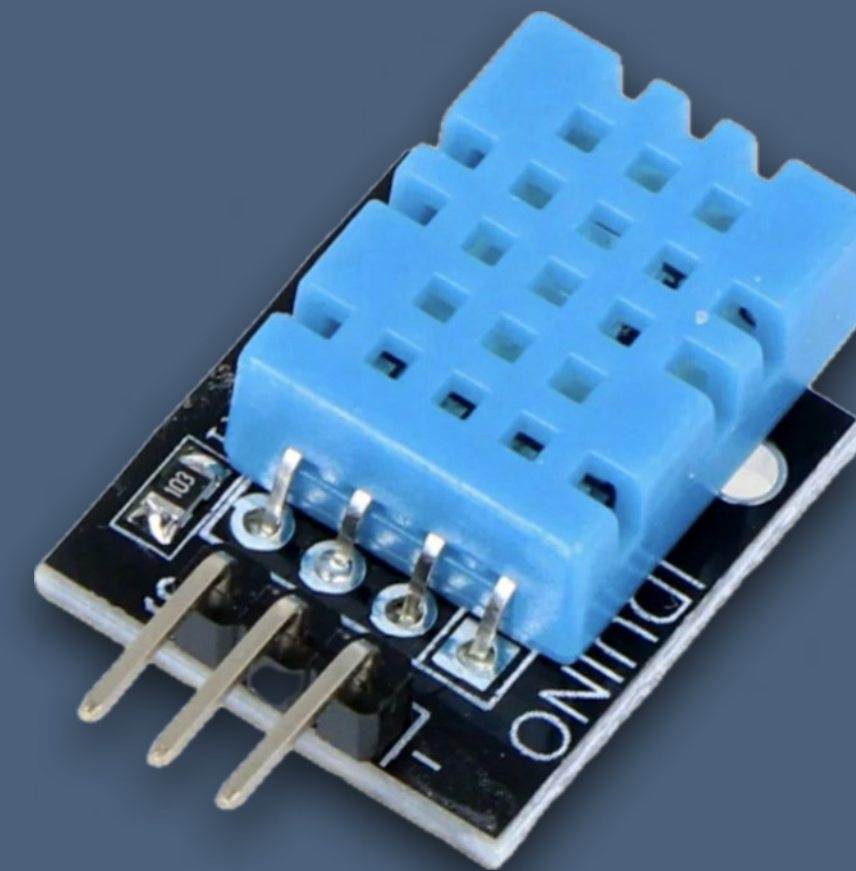


CALLIOPE mini
EXPERTS 2023



Keynotes

- Microcontroller und Pinout-Diagramm
- Programmiersprachen geeignete Editoren / Programmierumgebung
- Bibliotheken, Syntax und Code
- Sensoren und Displays
- Messwerterfassung und Auswertung
- Projekt Wetterstation



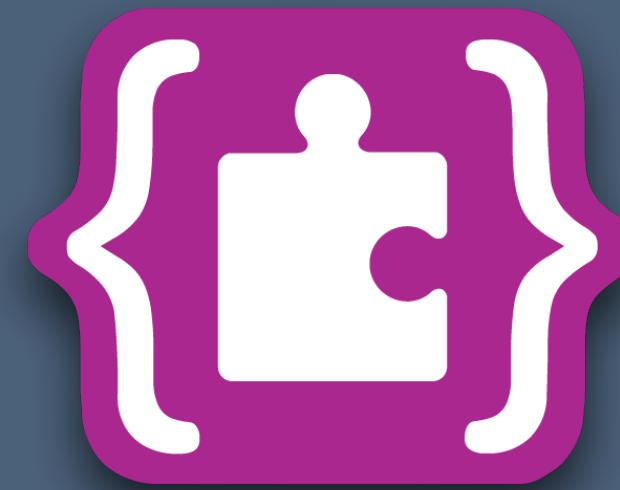
Integrated development environment (IDE)



Arduino IDE



MicroBlocks



pico
bricks



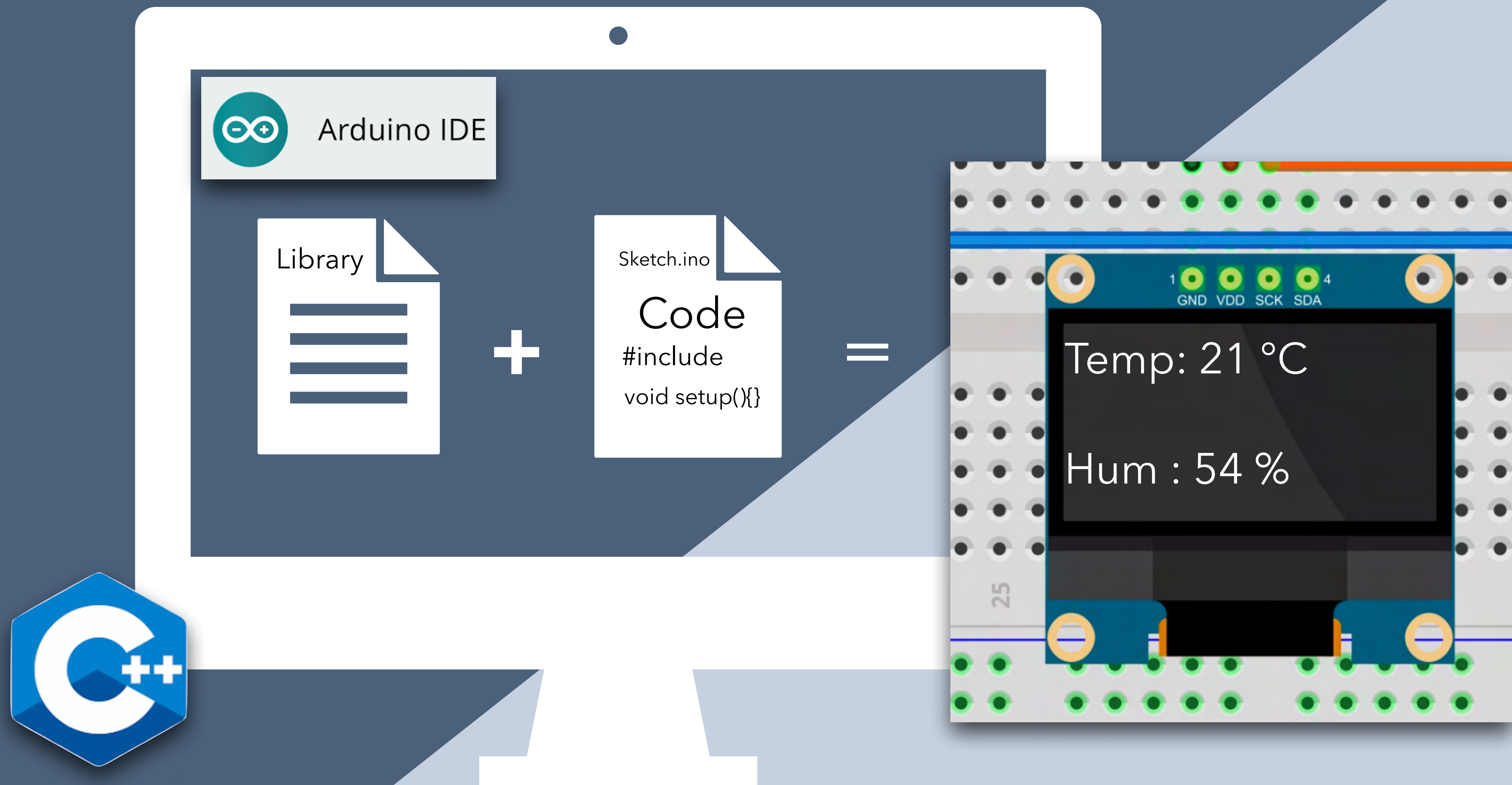
MicroPython



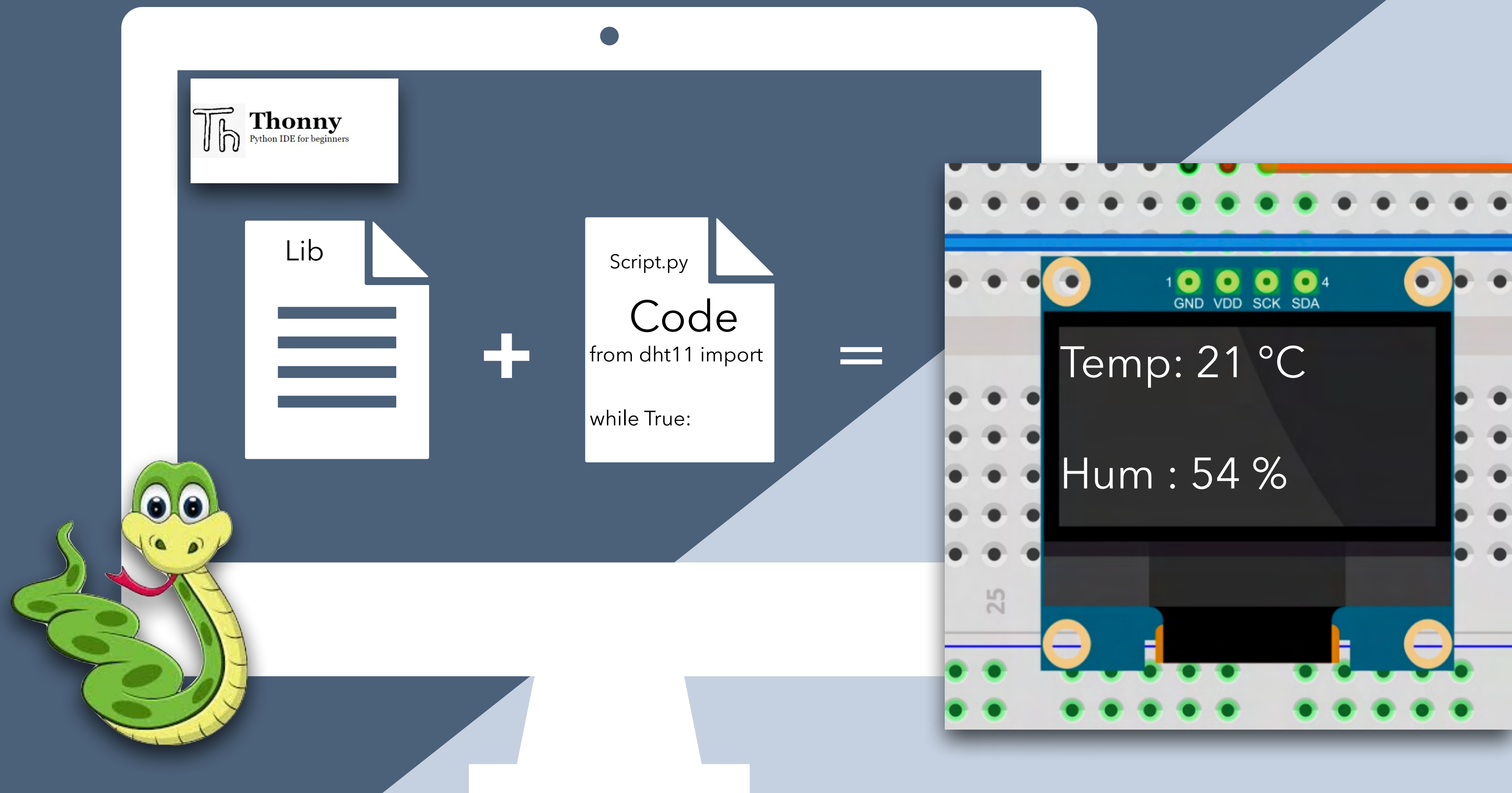
Visual Studio Code



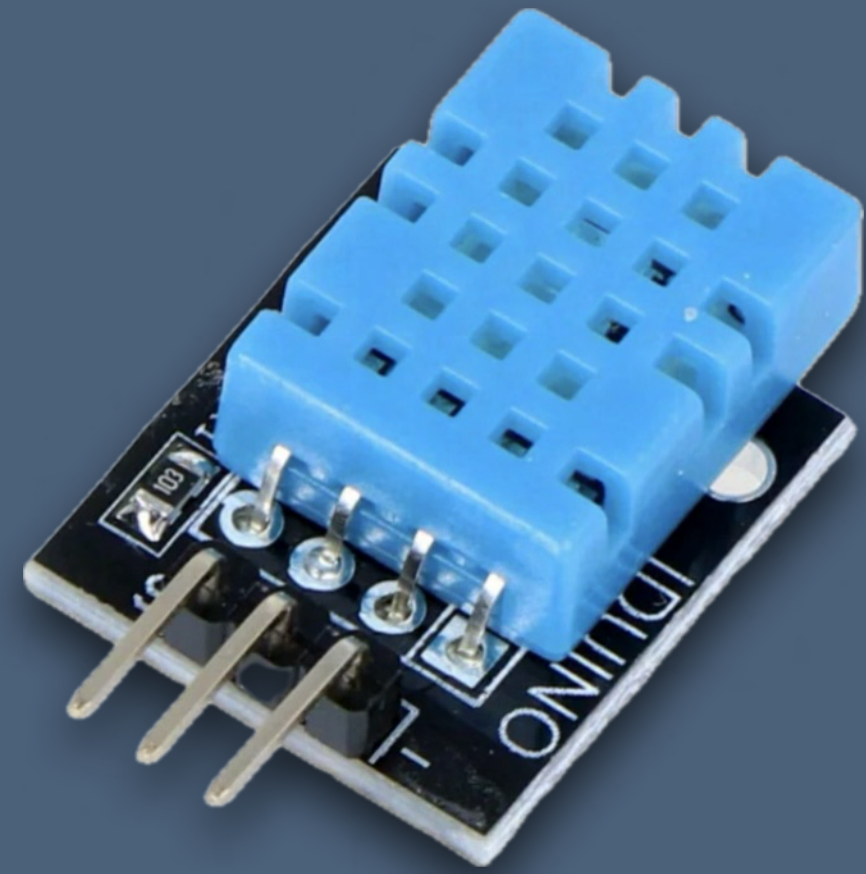
Integrated development environment (IDE)



Integrated development environment (IDE)



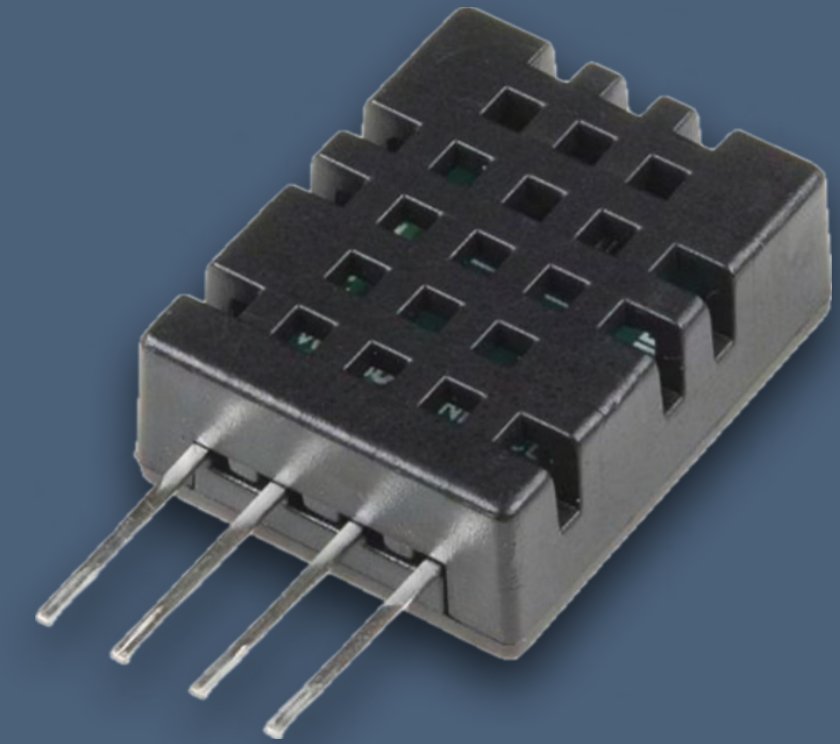
Sensoren und Displays



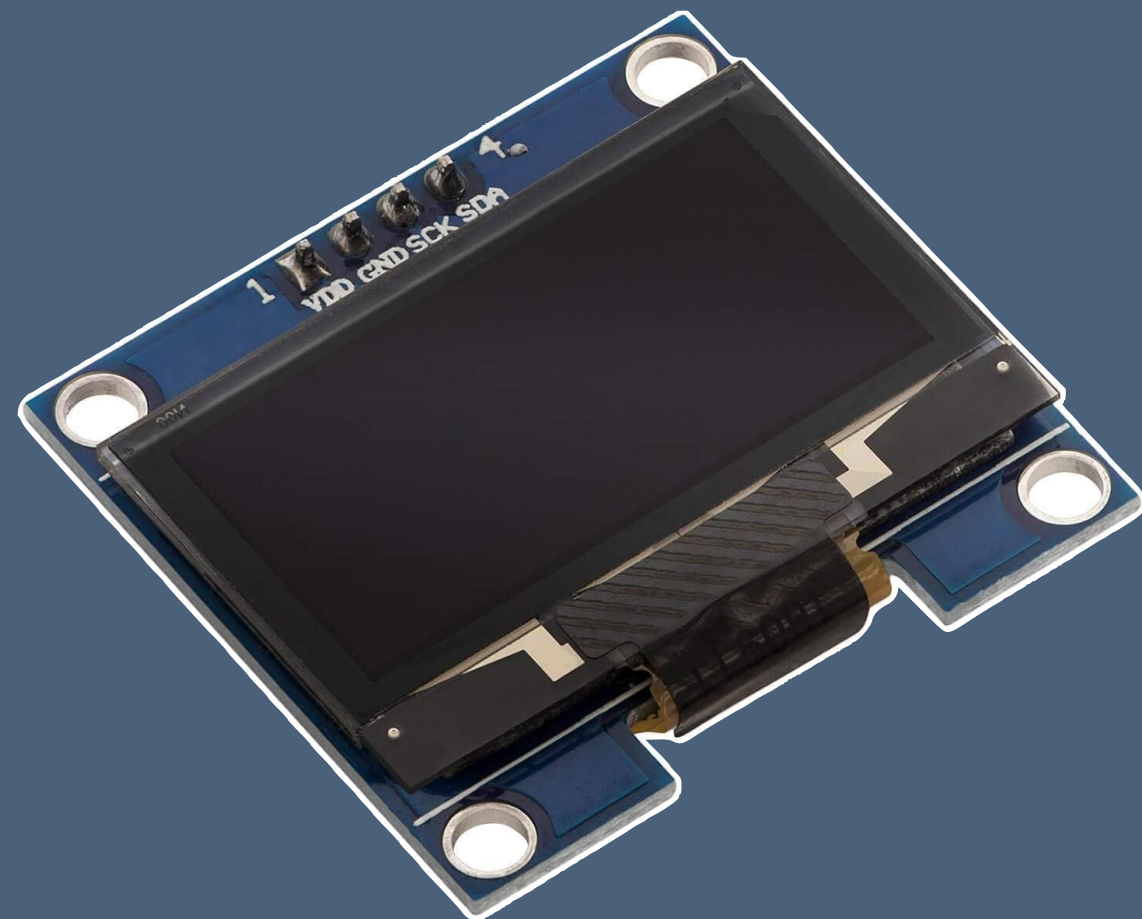
DHT11



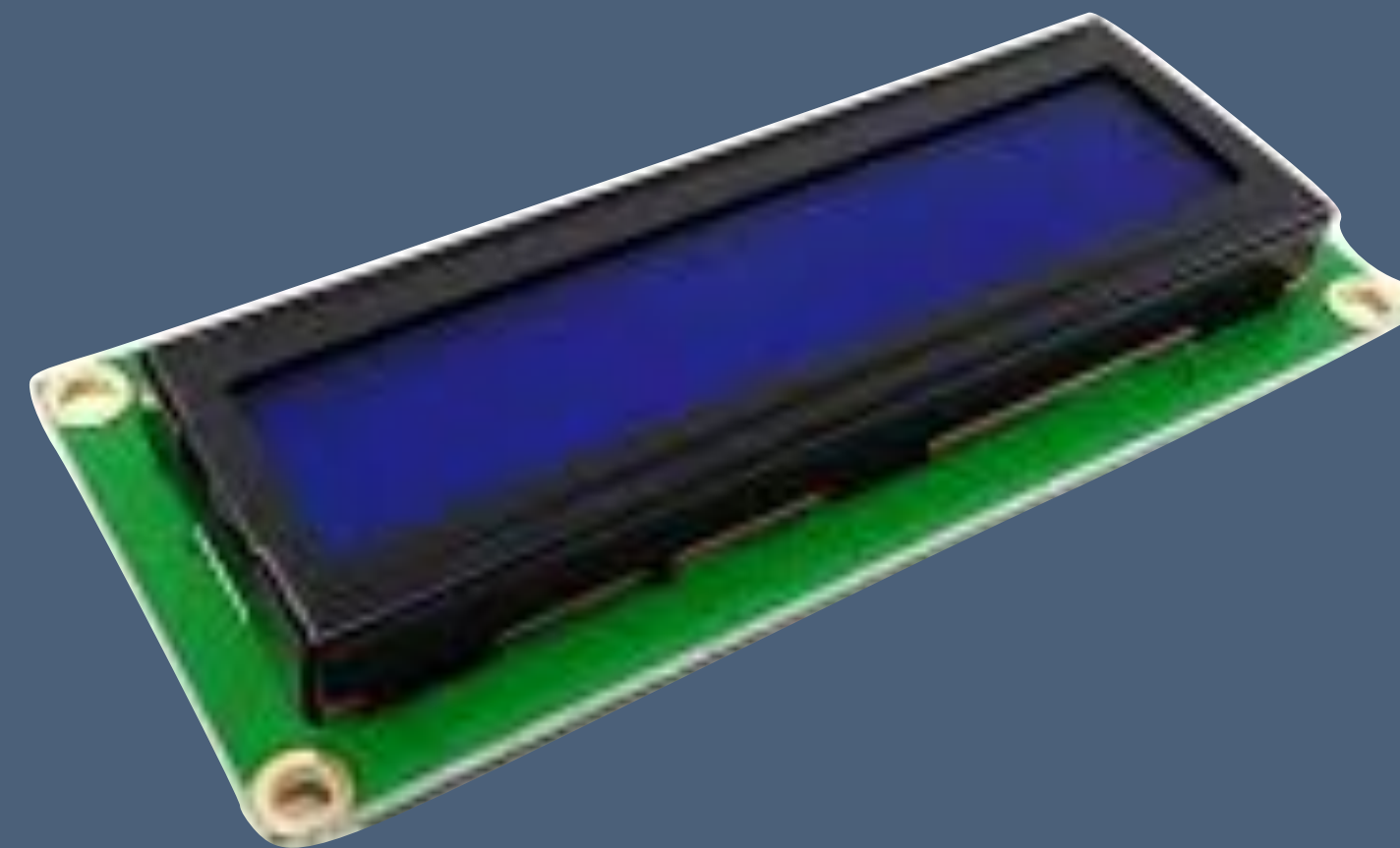
DHT22



DHT20

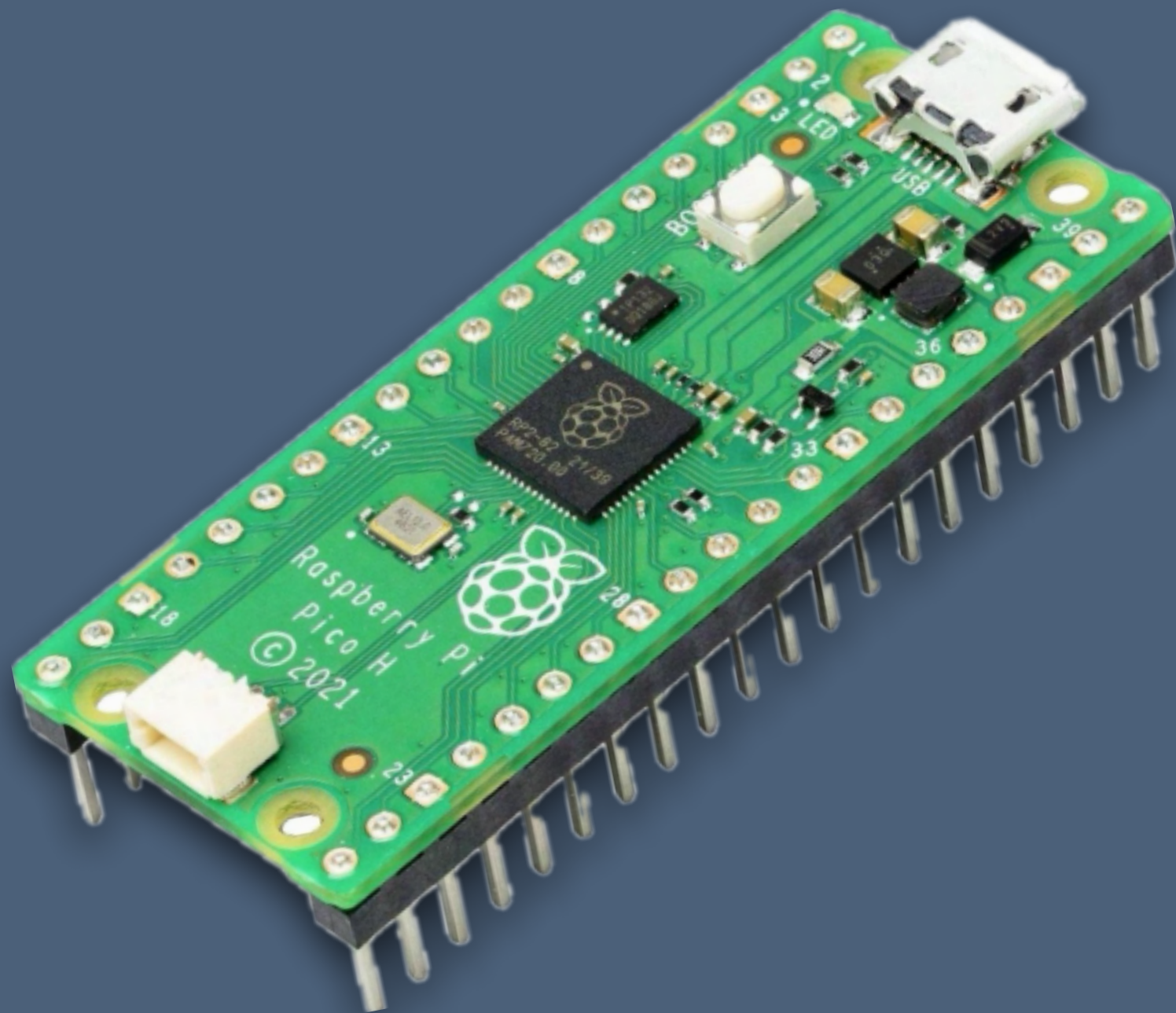


OLED SSD1306

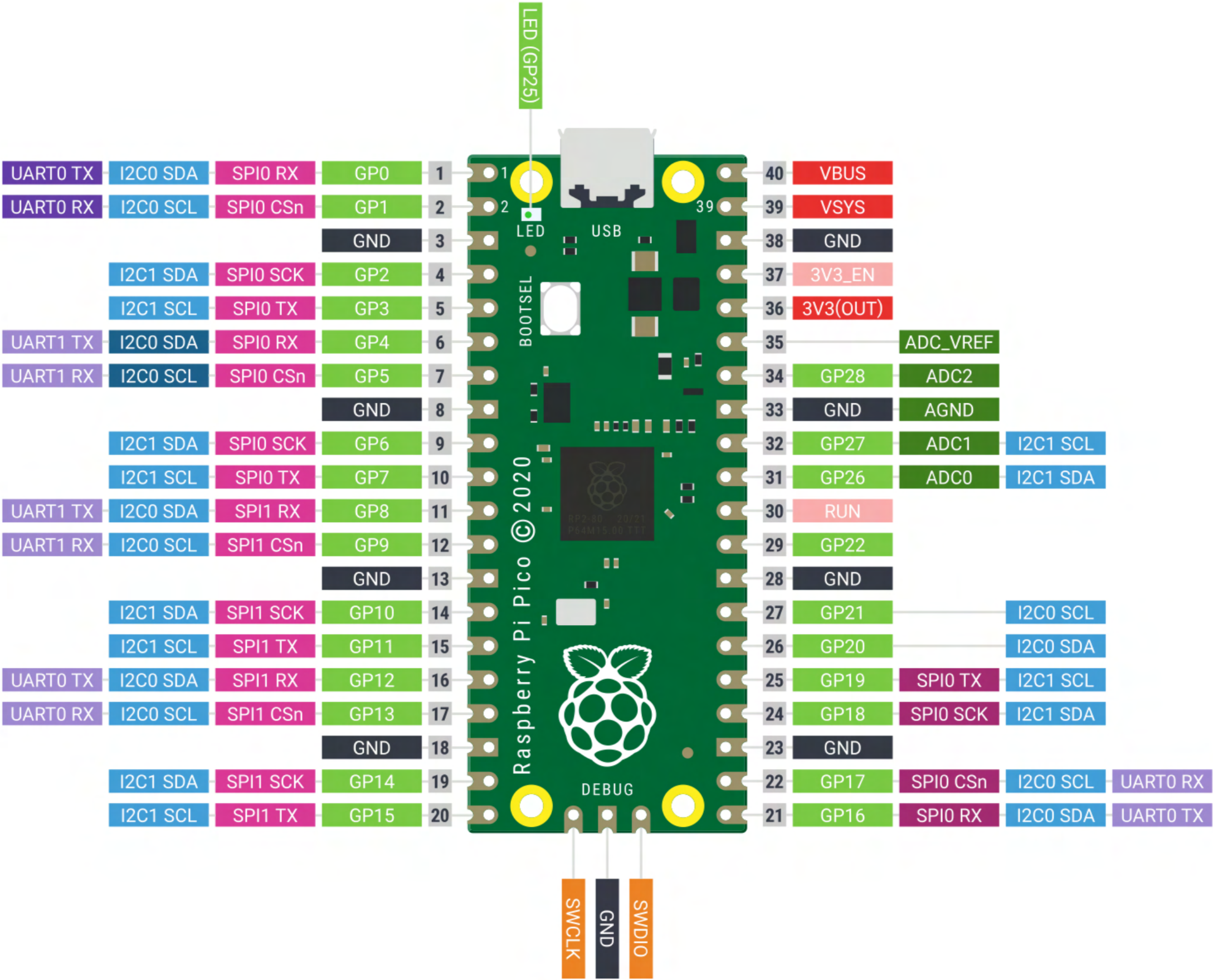


LCD 1602

Pinout Raspberry Pi Pico

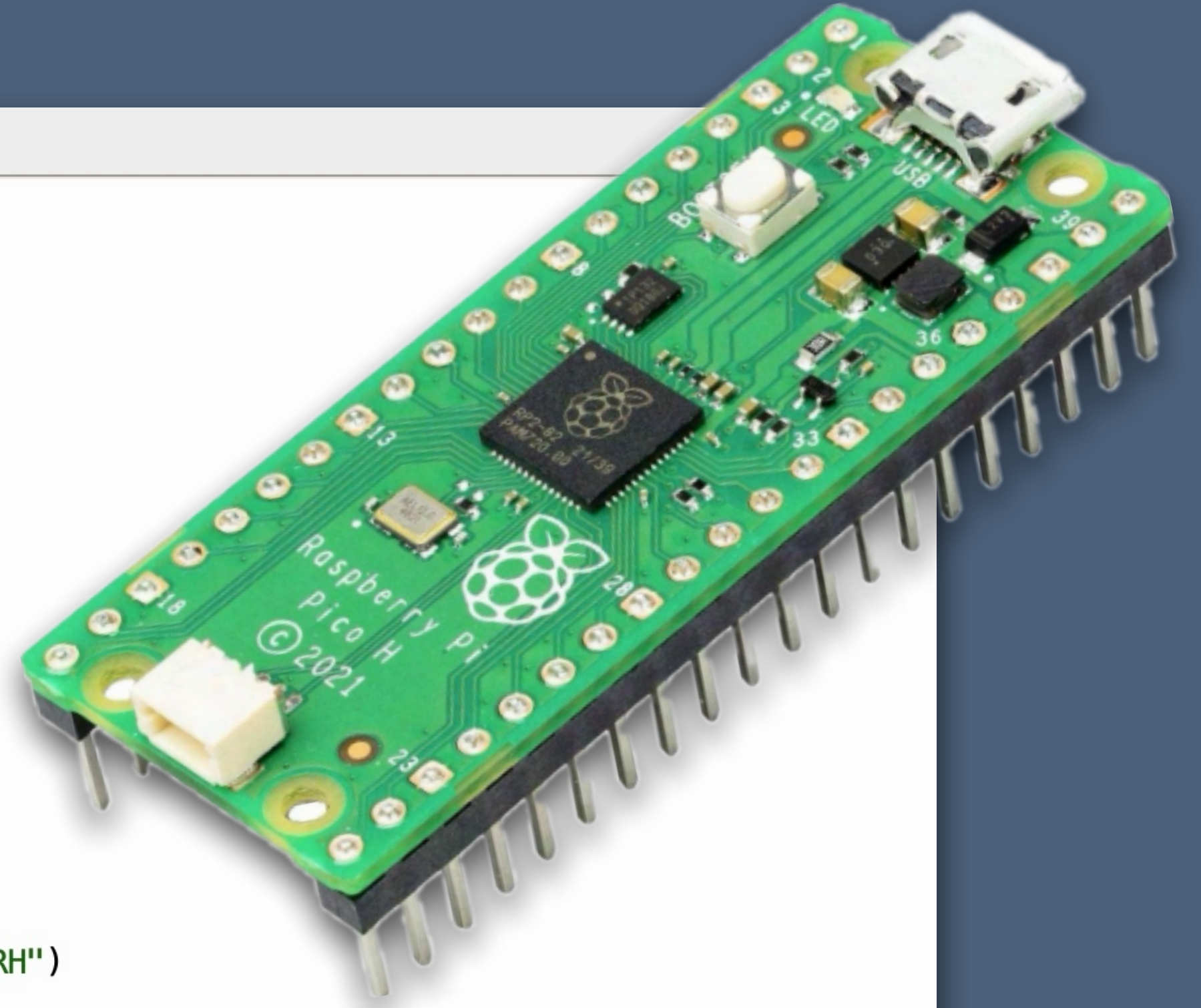


- Power
- Ground
- UART / UART (default)
- GPIO, PIO, and PWM
- ADC
- SPI / SPI (default)
- I2C / I2C (default)
- System Control
- Debugging

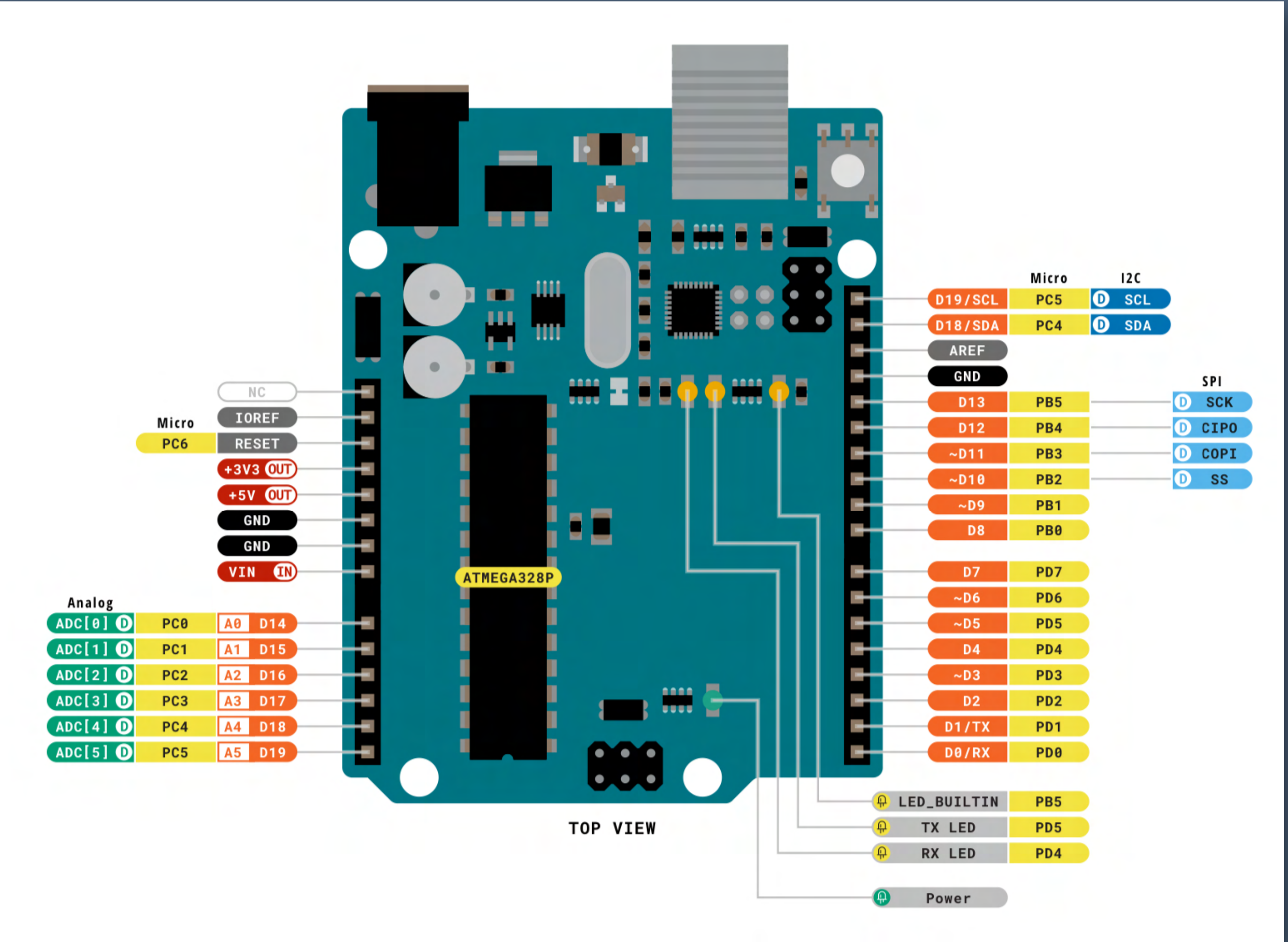
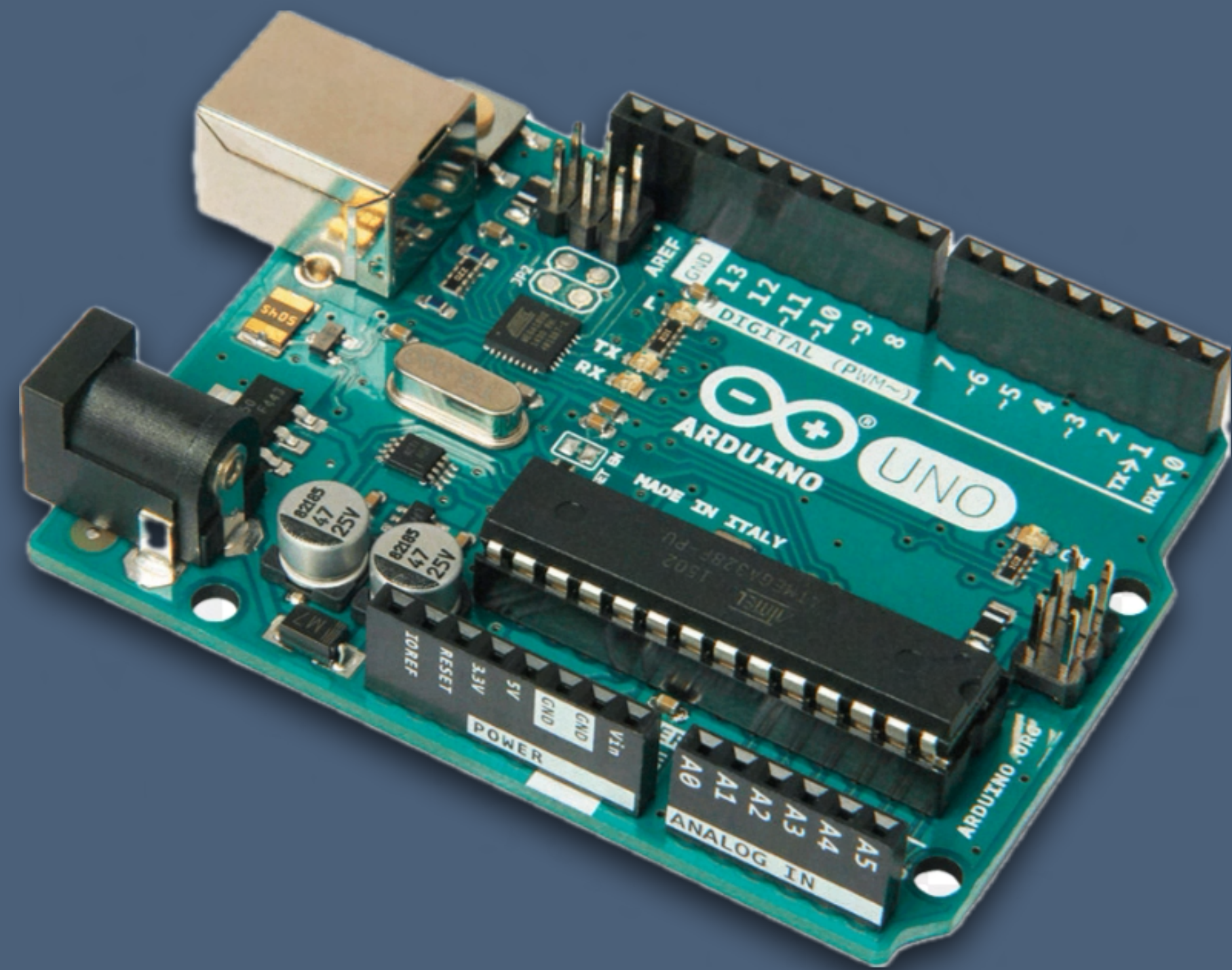


Raspberry Pi Pico

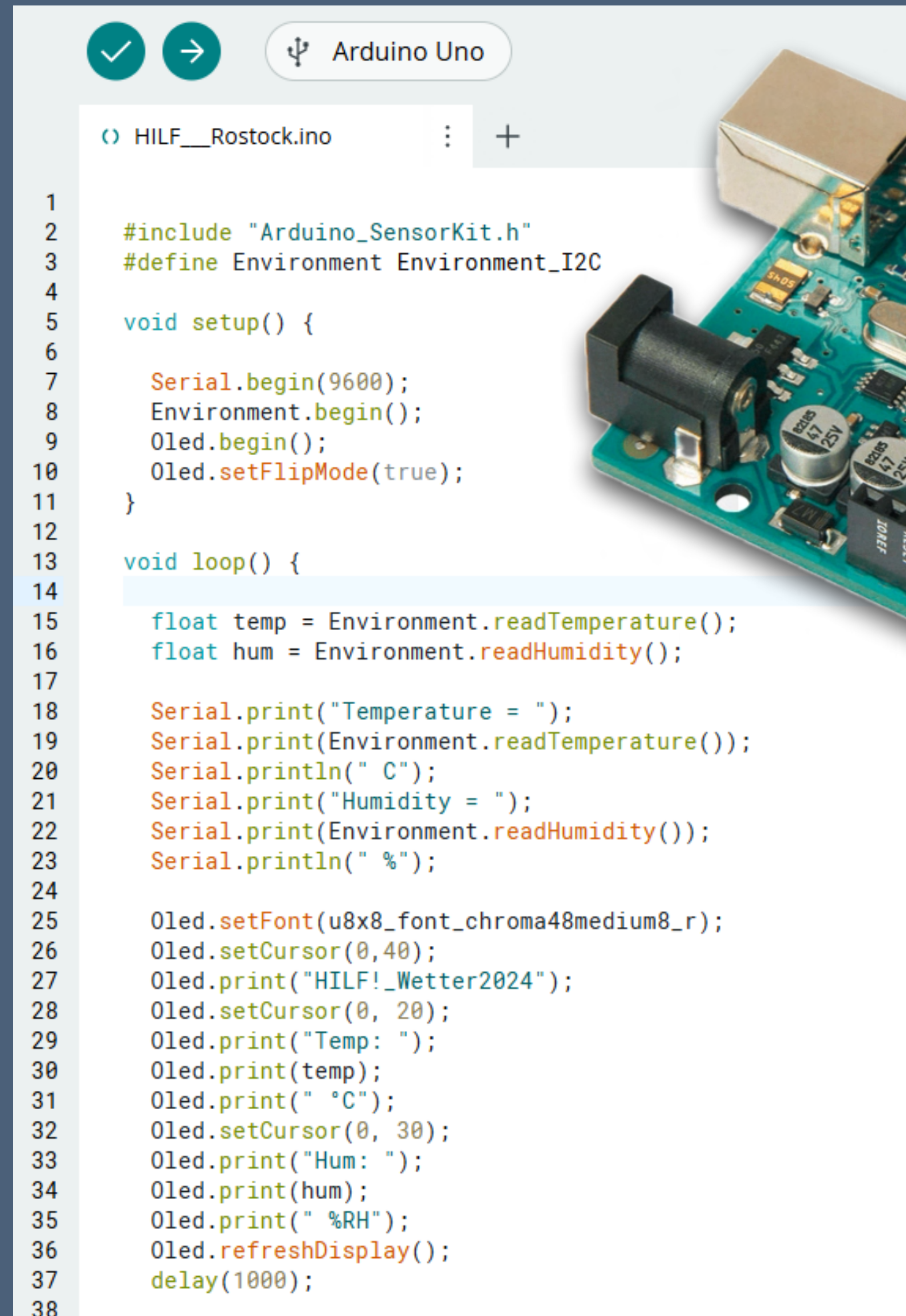
```
[ wetterstation.py ] * [ Pico_DHT.py ] [ helloworld.py ]  
1 from machine import Pin, SoftI2C  
2 from machine import Pin, I2C  
3 import ssd1306  
4 from dht20 import DHT20  
5 from utime import sleep  
6  
7 i2c0_sda = Pin(8)  
8 i2c0_scl = Pin(9)  
9 i2c0 = I2C(0, sda=i2c0_sda, scl=i2c0_scl, freq= 400000)  
10  
11 dht20 = DHT20(0x38, i2c0)  
12  
13 oled_width = 128  
14 oled_height = 64  
15 oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c0)  
16  
17 while True:  
18     measurements = dht20.measurements  
19     temp = measurements['t']  
20     hum = measurements['rh']  
21     print(f"Temperature: {measurements['t']} °C, humidity: {measurements['rh']} %RH")  
22     sleep(1)  
23     oled.fill(0)  
24     oled.text('Wetterstation', 0, 0)  
25     oled.text('HILF! 2024', 0, 10)  
26     oled.text(f"Temp: {measurements['t']} C", 0, 20)  
27     oled.text(f"Hum: {measurements['rh']} %RH", 0, 30)  
28     oled.text('Hum: ', 0, 40)  
29     oled.text(str(round(hum, 1)), 50, 40)  
30     oled.text('Temp: ', 0, 50)  
31     oled.text(' %RH ', 85, 40)  
32     oled.text(str(round(temp, 1)), 50, 50)  
33     oled.text(' C ', 85, 50)  
34     oled.show()  
35     sleep(1)
```



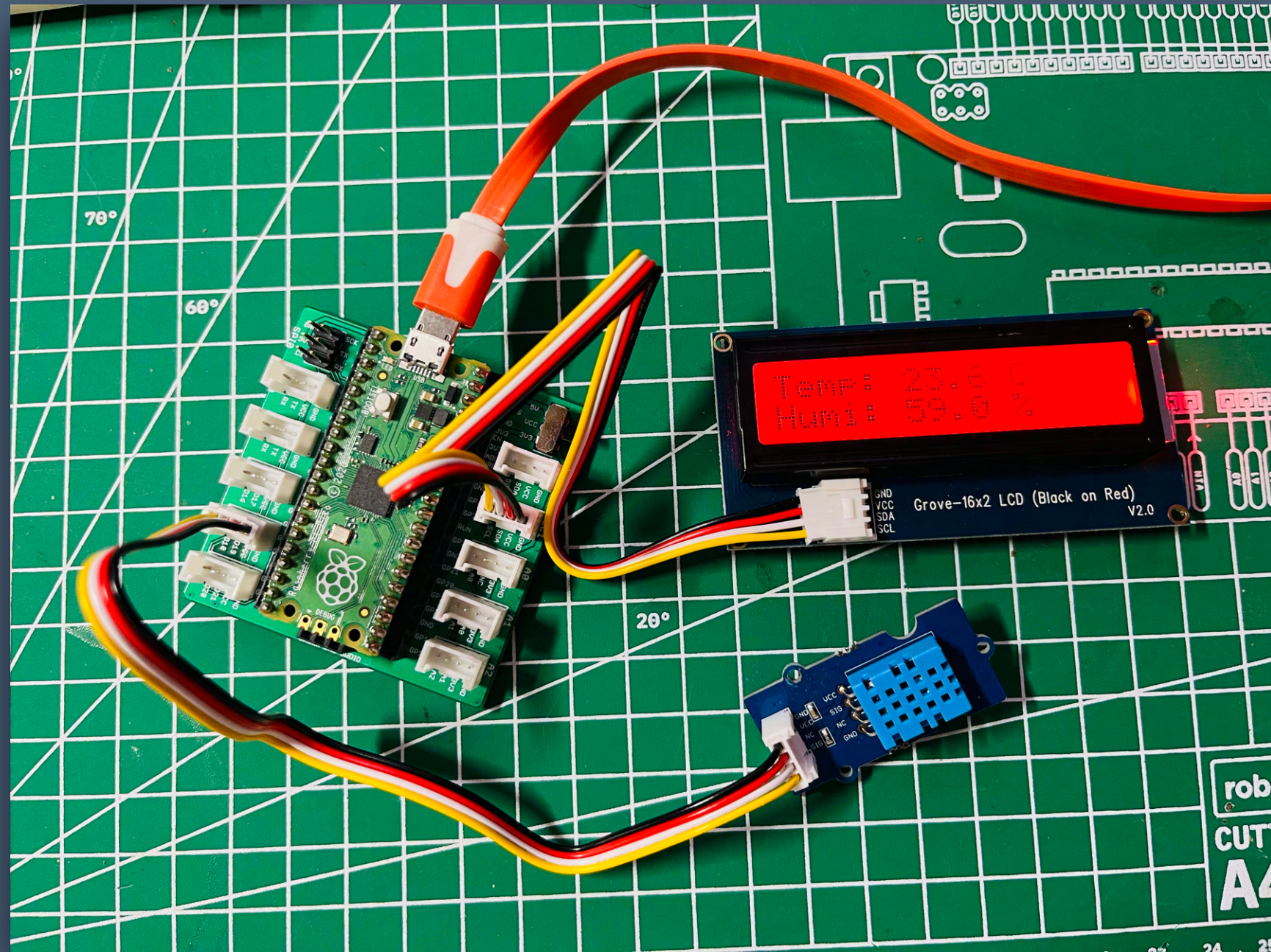
Pinout Arduino UNO R3



Arduino UNO R3

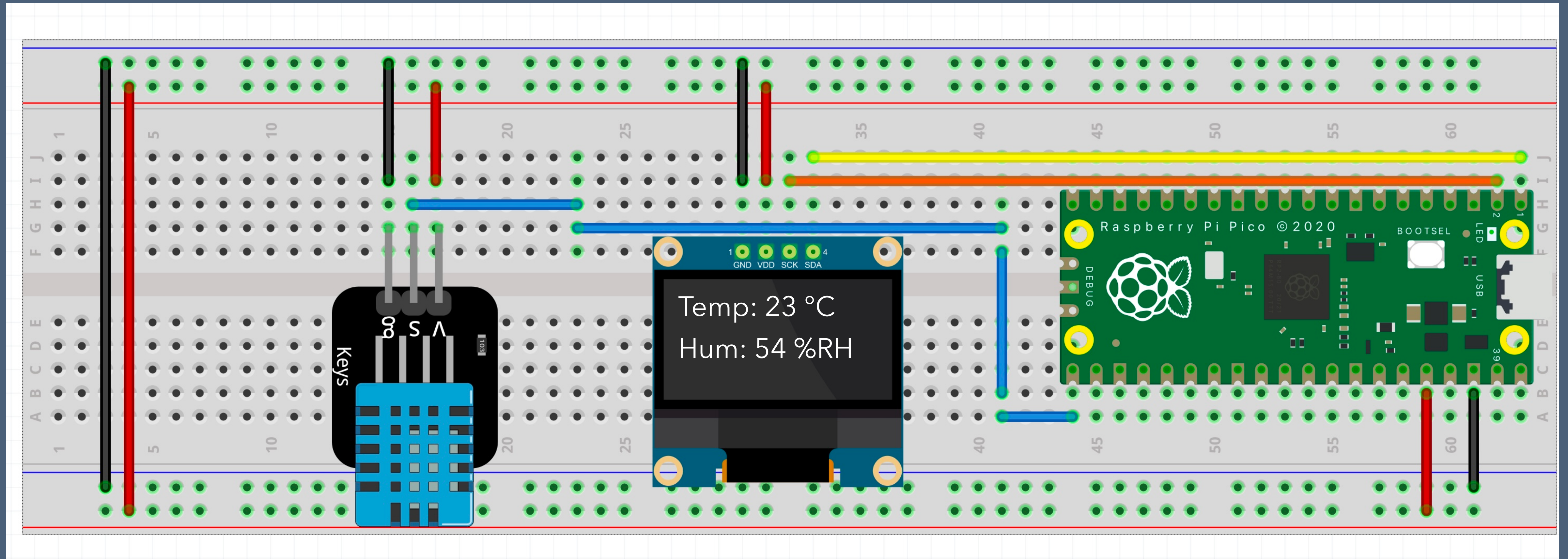


Pico Seeed Grove Shield



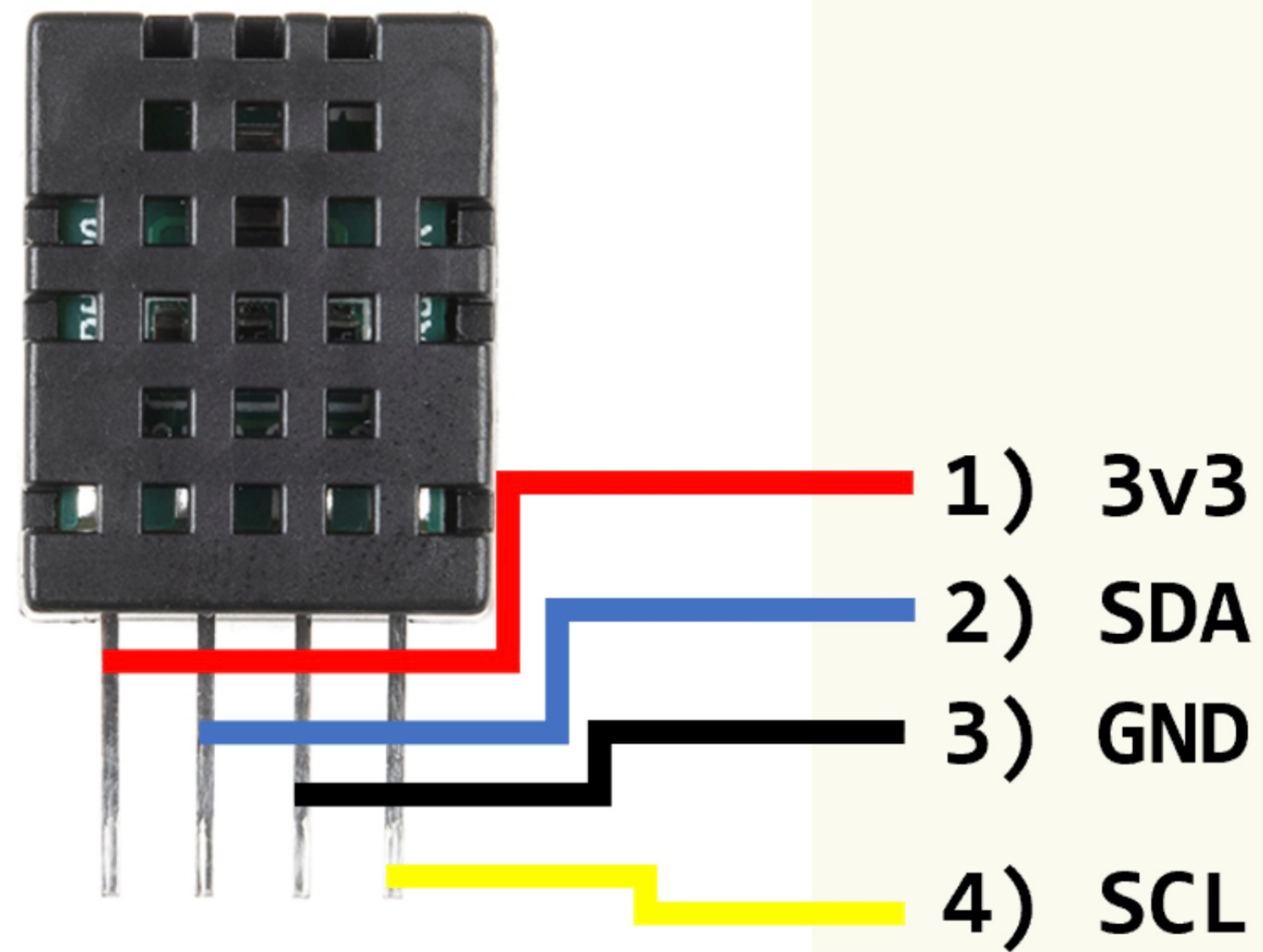
```
1 from dht11 import *
2 from machine import Pin, I2C
3 from lcd1602 import LCD1602
4 from time import sleep
5
6 i2c = I2C(1, scl=Pin(7), sda=Pin(6), freq=400000)
7 d = LCD1602(i2c, 2, 16)
8
9 dht2 = DHT(18)
10
11
12
13 while True:
14     temp, humid = dht2.readTempHumid()
15     d.home()
16     d.print("Temp: " + str(temp))
17     d.print(" C")
18     d.setCursor(0, 2)
19     d.print("Humid: " + str(humid))
20     d.print(" %")
21     sleep(0.5)
```

Breadboard, Pico+DHT11+SSD1306 OLED

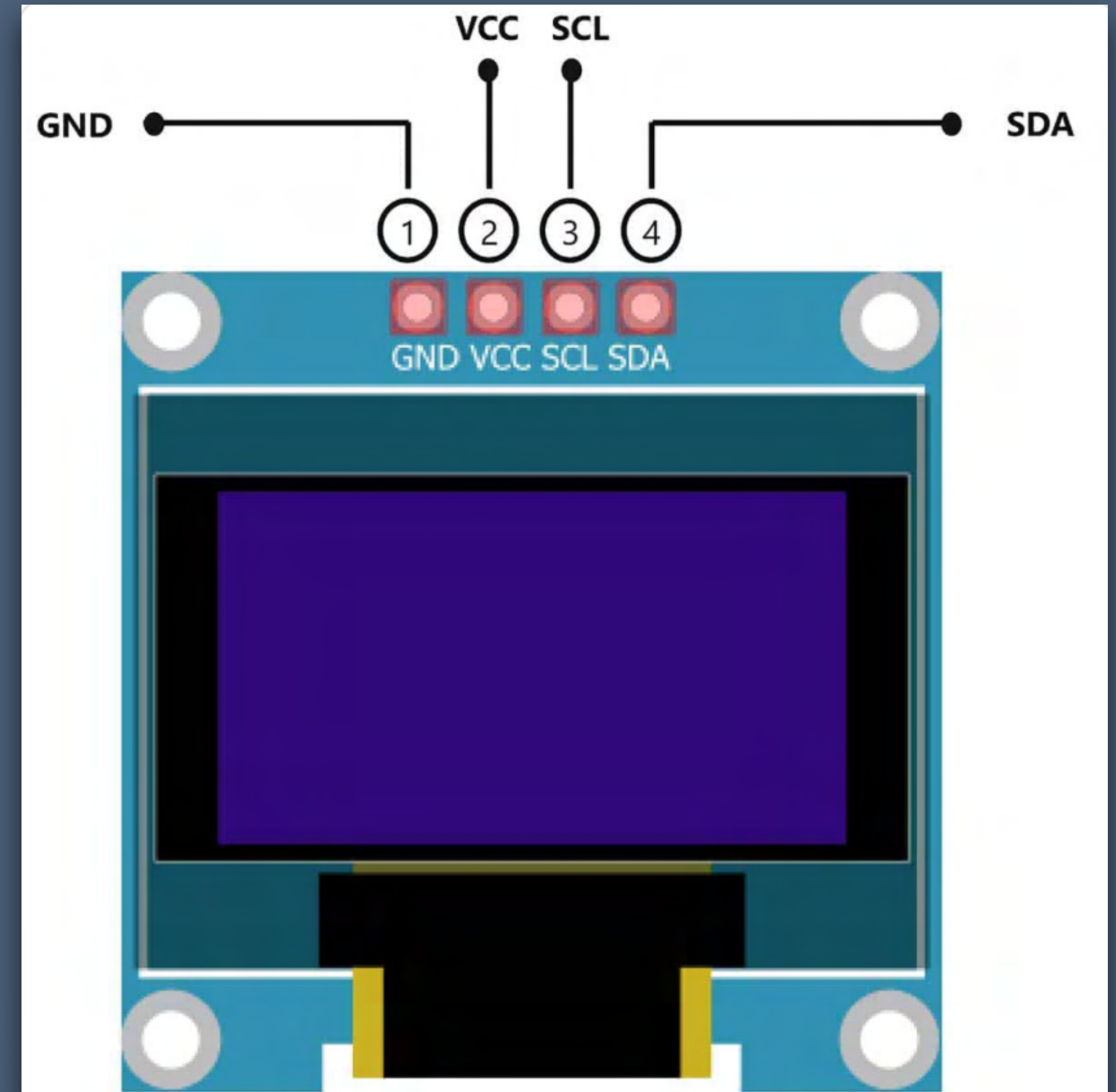


Pinout, Pico+DHT20+SSD1306 OLED

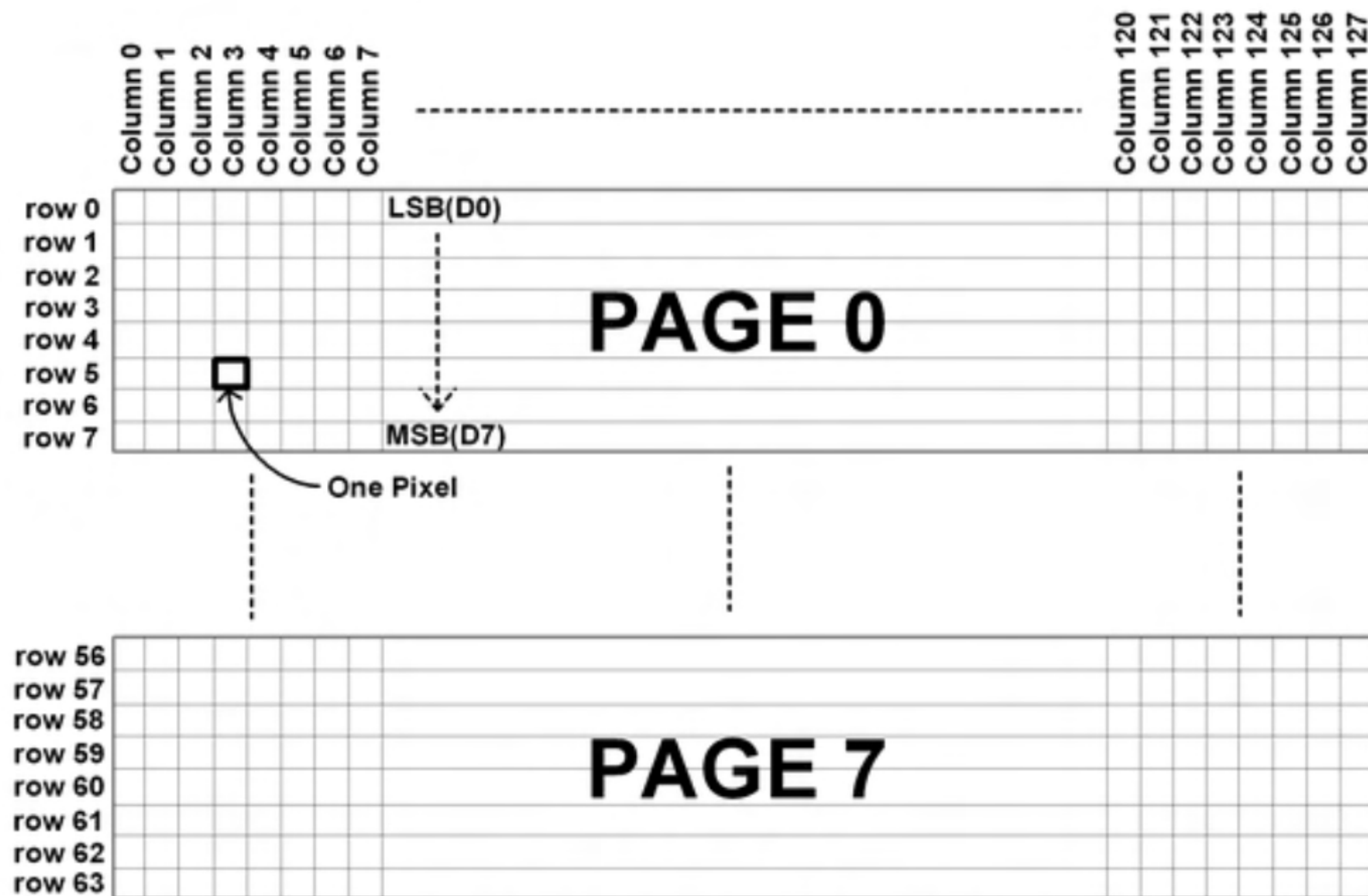
Wiring Guide - 4 Pins



1	GND	Connect the ground of the supply to this pin.
2	VCC	There is a power supply pin here. A supply of 3.3V or 5V can be provided to this pin to power the display.
3	SCL (Serial Clock line)	It is a clock signal. Data is sent to other devices as clock ticks occur. Only the master generates the SCL.
4	SDA (Serial Data line)	Using SDA, slaves and masters can exchange data and acknowledgment.

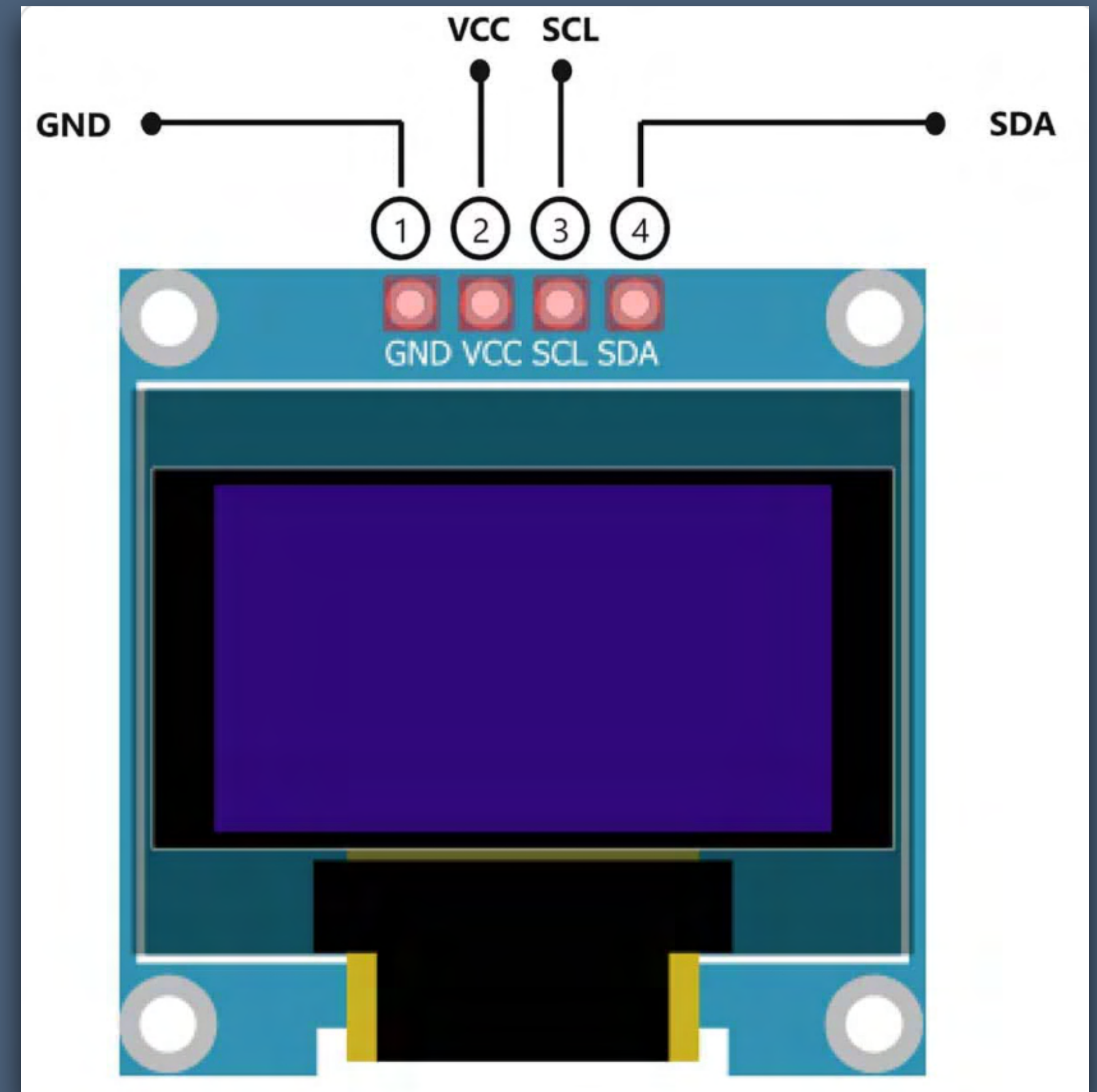


SSD1306 OLED

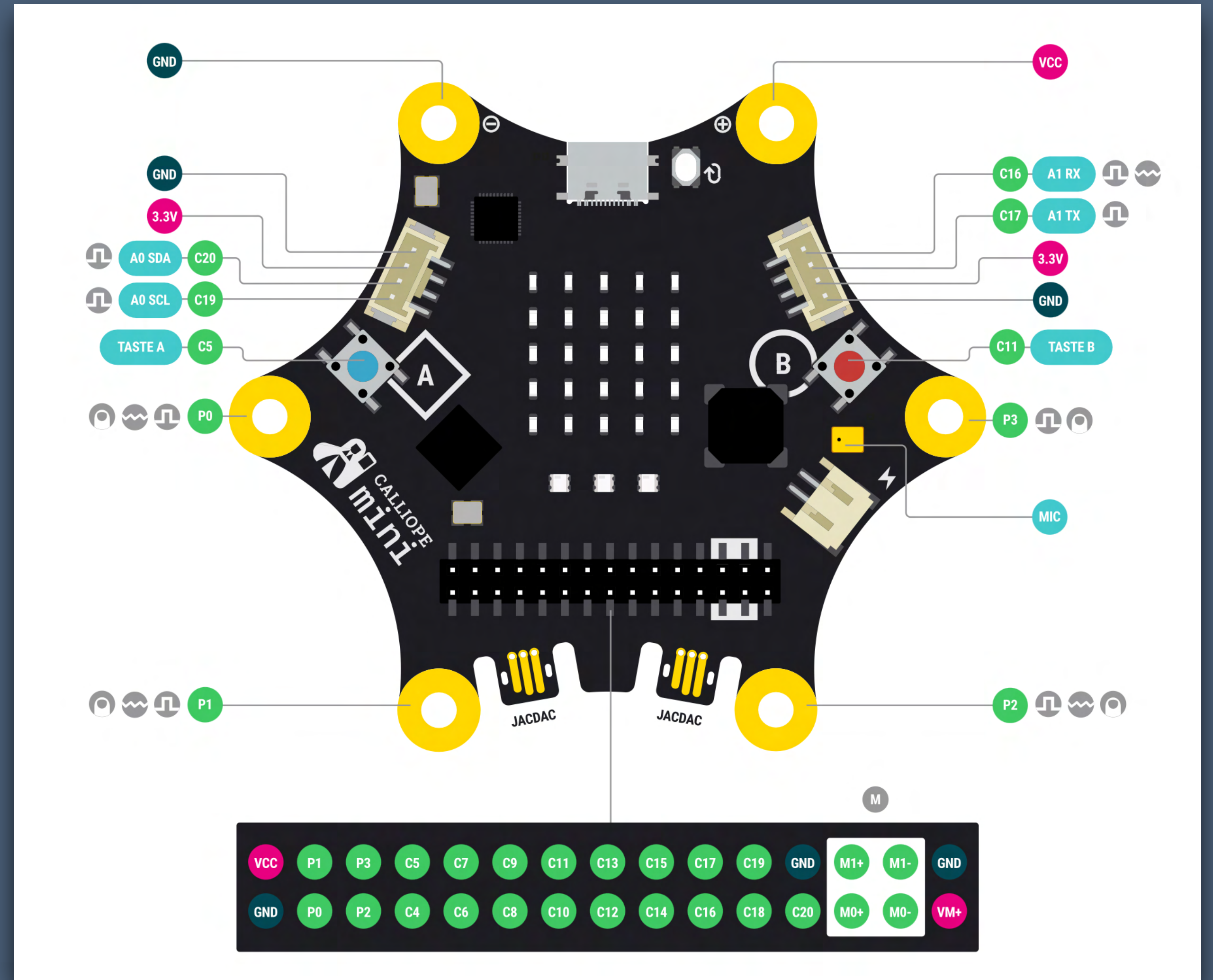
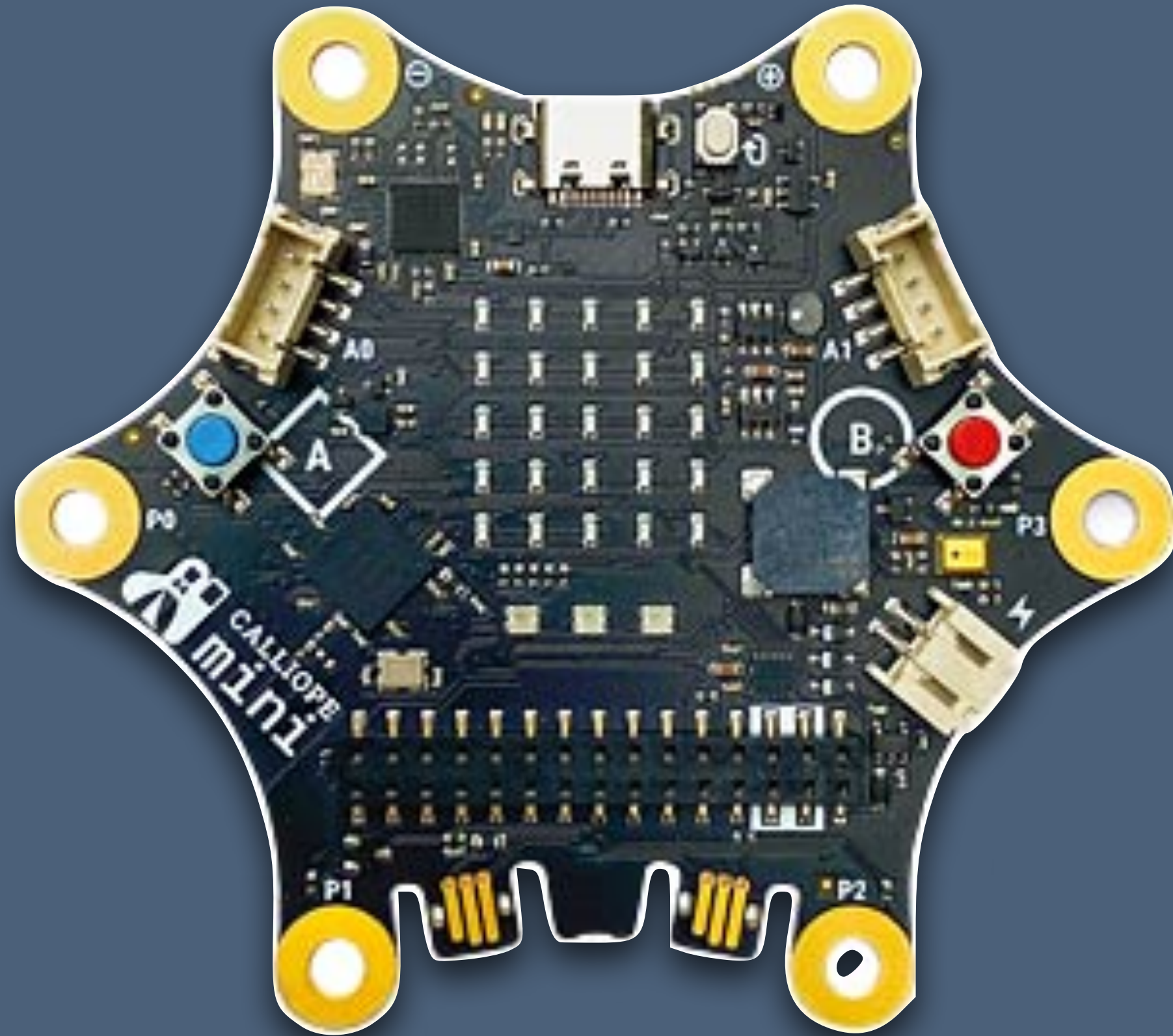


Pixelwise structure of OLED Display

128x64



Calliope Mini V3



<https://github.com/MKleinSB/pxt-dht20>

Calliope Mini V3 LCD-1602 Datalogger

Microsoft

CALLIOPE mini

Blöcke

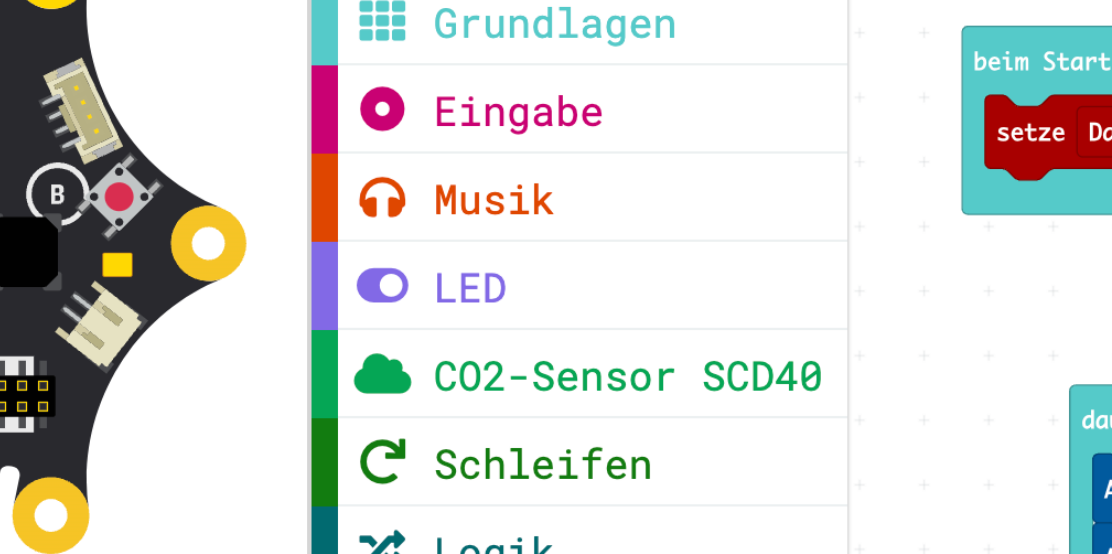
JavaScript

🏠

🔗

❓

⚙️



Suche...

Grundlagen

Eingabe

Musik

LED

C02-Sensor SCD40

Schleifen

Logik

Variablen

Mathematik

Datenlogger

HM3301

display

Motoren

Erweiterungen

Fortgeschritten

Daten anzeigen Simulator

Daten anzeigen Calliope mini

beim Start

setze Data auf 0

dauerhaft

Aufruf Anzeige Matrix

Aufruf Anzeige LCD

Aufruf Datalogger

Funktion Datalogger

ändere Data um 1

setze Zeitstempel nichts

Protokolldaten Spalte Data Wert Data

Spalte Temperatur Wert temperature °C

Spalte Luftfeuchtigkeit Wert relative humidity

Spalte CO2 Wert CO2

spiegle Daten seriell

pausiere (ms) 2000

Funktion Anzeige LCD

verbinde T:

wandle temperature °C um in Text

Schreibe C

H:

wandle relative humidity um in Text

auf Zeile 0 und bewege die Zeichen mit 1000 ms

Schreibe verbinde CO2: wandle CO2 um in Text ppm auf Zeile 1 und bewege die Zeichen mit 1000 ms

bei voller Logdatei

lösche Protokoll schnell

Funktion Anzeige Matrix

wenn CO2 > 2000 dann

zeige Symbol

setze Farbe auf Rot 255 Grün 0 Blau 0

sonst wenn CO2 > 1000 und CO2 < 2000 dann

zeige Symbol

setze Farbe auf Rot 255 Grün 255 Blau 0

sonst wenn CO2 < 1000 dann

zeige Symbol

setze Farbe auf Rot 0 Grün 255 Blau 0

Herunterladen

Luftqualität

📄

🔄

🏠

🔗

❓

⚙️

Calliope Mini V2 SSD1306

Grundlagen

- Eingabe
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- SSD1306 OLED
- DHT20
- Motoren
- Erweiterungen
- Fortgeschritten
- Funktionen
- Arrays

dauerhaft

- setze hum auf DHT20 Sensor Feuchtigkeit (0~100) %
- setze temp auf DHT20 Sensor Temperatur (°C)
- Aufruf OLED_Wetter
- Aufruf Wetter_LED_matrix

beim Start

- Display anschalten
- initialisiere Display
- drehe Display

Funktion Wetter_LED_matrix

- wenn temp $\geq$ 25 dann
 - zeige Symbol [LED Matrix Icon]
 - setze Farbe auf [Red]
- sonst wenn temp $<$ 25 dann
 - zeige Symbol [LED Matrix Icon]
 - setze Farbe auf [Green]

Funktion OLED_Wetter

- setze Cursor auf Zeile 1 und Spalte 0
- schreibe "HILF!_Wetter2024" auf das Display
- setze Cursor auf Zeile 3 und Spalte 0
- schreibe verbinde "Temp:" runden temp "C" auf das Display
- setze Cursor auf Zeile 5 und Spalte 0
- schreibe verbinde "Hum:" runden hum "%RH" auf das Display
- pausiere (ms) 200

Follow me on social media



@r_schulz_maker



r_schulz_maker



@r_schulz_maker@mastodon.social

