84X48 LCD DISPLAY

SBC-LCD84x48

1. GENERAL INFORMATION

Dear Customer,
thank you for choosing our product. In the following, we will show you what to pay attention to during commissioning and use.

If you encounter any unexpected problems during use, please feel free to contact us.

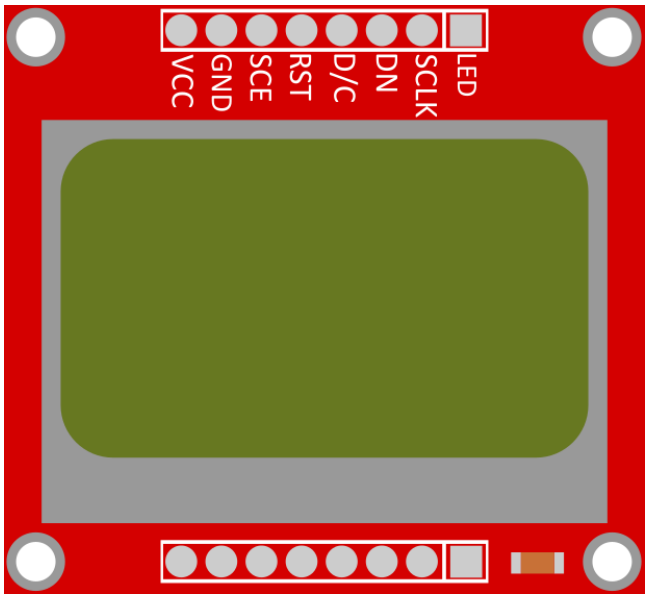
The display can be used with many common developer boards such as Raspberry Pi or Arduino, for example, to display measured values or measured value graphics directly on the microcontroller or single-board computer.

2. DIFFERENCES BETWEEN THE REVISIONS

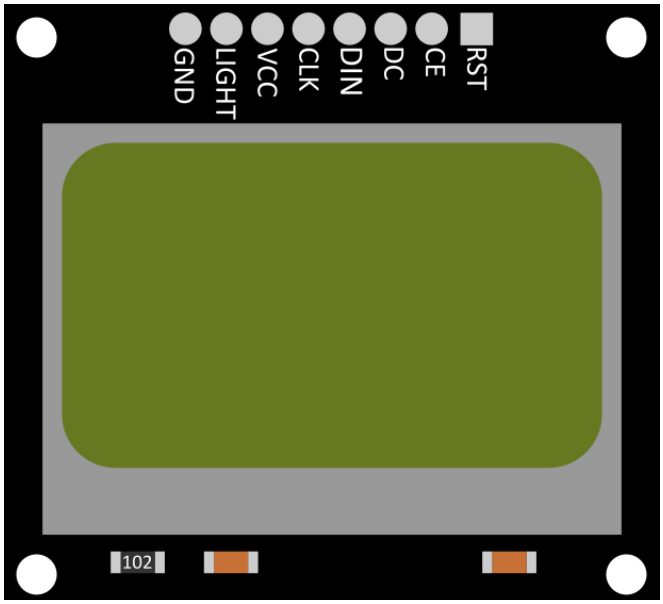
This product has a new revision that replaces the previous revision. These changes include the pin out, the LED and the silkscreen.

Below you can see 2 drawings of the different revisions.

Revision 1



Revision 2

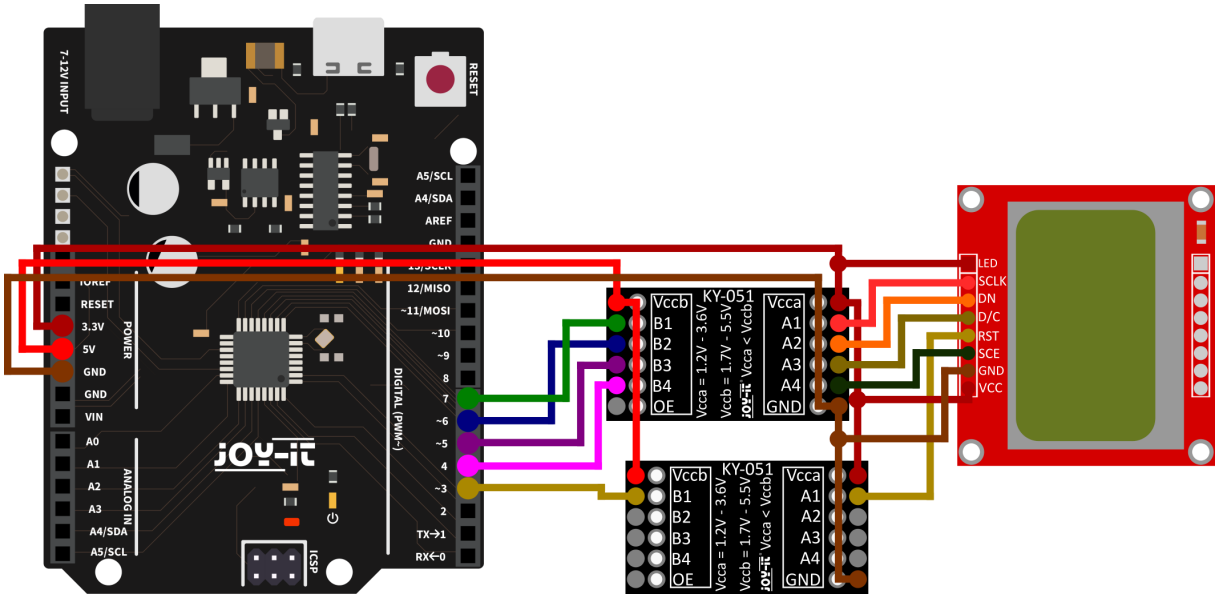


3. USE WITH AN ARDUINO

3.1 Connecting the display (REV 1)

Connect the display as shown in the following picture or table to the pins of the "A-side" of your Voltage Translator and then from the "B-side" of the Voltage Translator to your Arduino.
Here we recommend the **KY-051 Voltage Translator** from **Joy-IT**.

> It is necessary to use a logic level converter, because the display has a 3.3 V logic level, while the Arduino has a 5 V logic level. Not doing so may cause damage to the display.

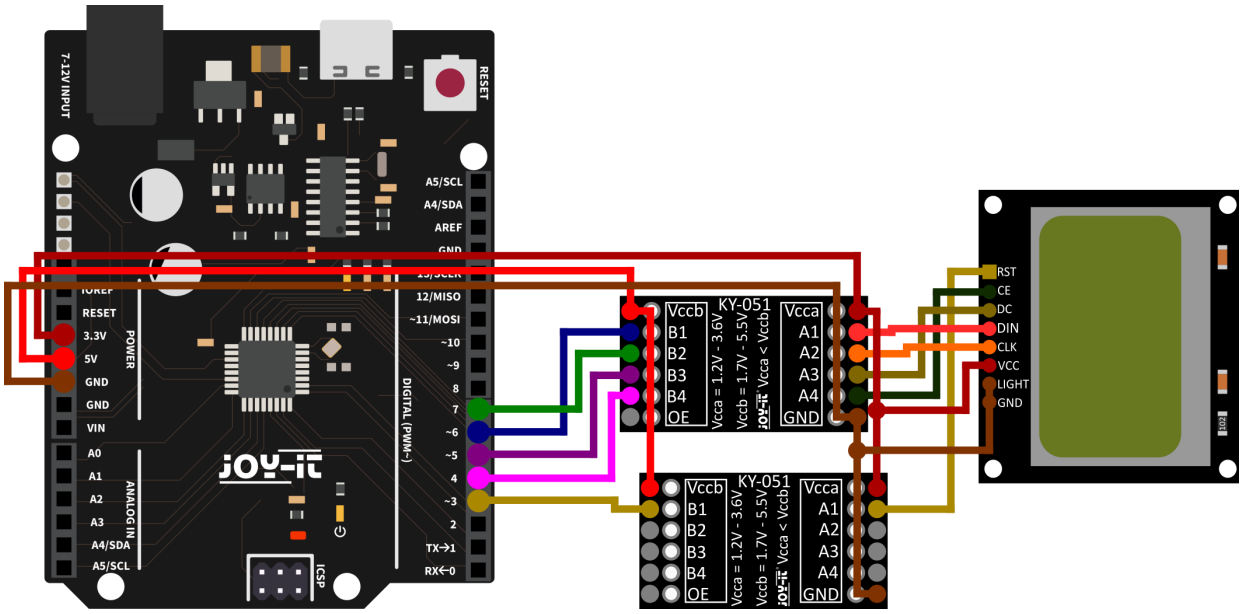


LCD	Arduino	KY-051 (1)	KY-051 (2)
VCC	3V3	Vcca	Vcca
-	5V	Vccb	Vccb
GND	GND	GND	GND
SCE	-	A4	-
-	D4	B4	-
RST	-	-	A1
-	D3	-	B1
D/C	-	A3	-
-	D5	B3	
DN	-	A2	-
-	D6	B2	-
SCLK	-	A1	-
-	D7	B1	-

3.2 Connecting the display (REV 2)

Connect the display as shown in the following picture or table to the pins of the "A-side" of your Voltage Translator and then from the "B-side" of the Voltage Translator to your Arduino.
Here we recommend the **KY-051 Voltage Translator** from **Joy-IT**.

> It is necessary to use a logic level converter, because the display has a 3.3 V logic level, while the Arduino has a 5 V logic level. Not doing so may cause damage to the display.



LCD	Arduino	KY-051 (1)	KY-051 (2)
GND	GND	GND	GND
LIGHT	GND	-	-
VCC	3V3	Vcca	Vcca
-	5V	Vccb	Vccb
CLK	-	A2	-
-	D7	B2	-
DIN	-	A1	-
-	D6	B1	-
DC	-	A3	-
-	D5	B3	
CE	-	A4	-
-	D4	B4	-
RST	-	-	A1
-	D3	-	B1

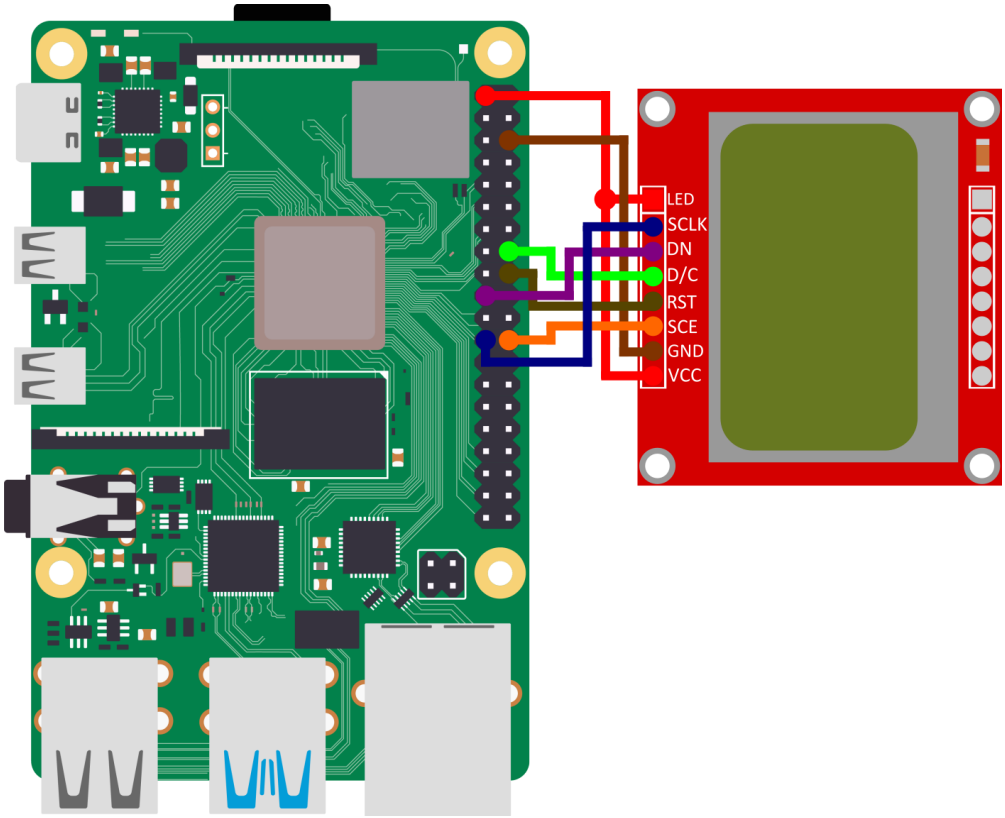
3.3 Code example Arduino

This example uses the Adafruit **PCD8544-Nokia-5110-LCD-library** library, which is released by **Adafruit** under the **BSD license**. After installing this library in your Arduino IDE, you can find a code example at **File → Examples → Adafruit PCD8544 Nokia 5110 LCD Library → pcdtest**. You can upload this code to your Arduino using Upload. Also make sure that the correct settings are made under **Tools → Board** and **Tools → Port**.

4. USE WITH A RASPBERRY PI

4.1 Connecting the display (REV 1)

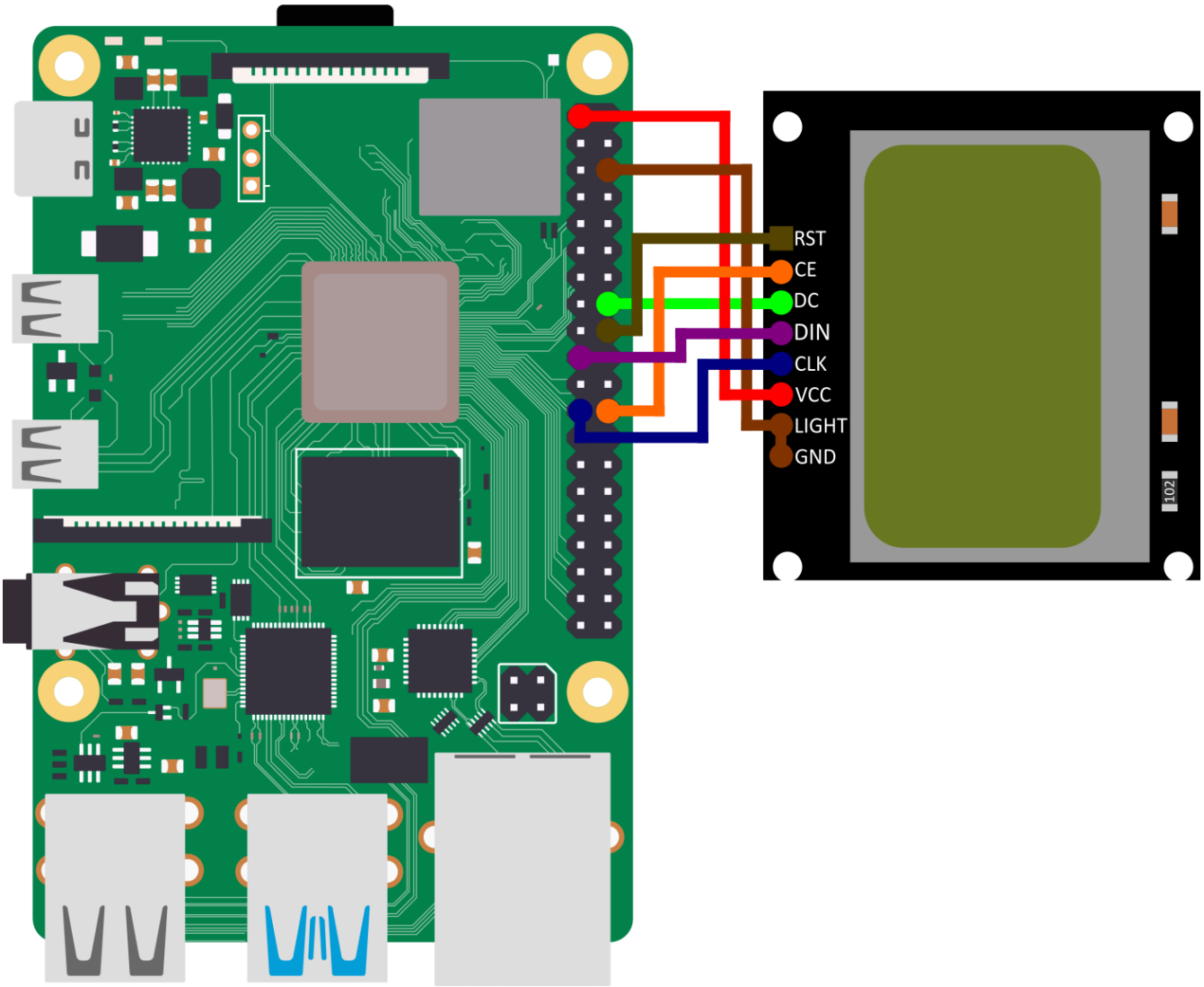
Connect the display to the pins of your Raspberry Pi as shown in the following picture or table.



Raspberry Pi	84 x 48 LCD
Pin 1 (3.3 V)	8 - LED
Pin 23 (BCM 11 / SCLK)	7 - SCLK
Pin 19 (BCM 10 / MOSI)	6 - DN < MOSI >
Pin 16 (BCM 23)	5 - D/C
Pin 18 (BCM 24)	4 - RST
Pin 24 (BCM 8)	3 - SCE
Pin 6 (GND)	2 - GND
Pin 1 (3.3 V)	1 - VCC

4.2 Connecting the display (REV 2)

Connect the display to the pins of your Raspberry Pi as shown in the following picture or table.



Raspberry Pi	84 x 48 LCD
Pin 18 (BCM 24)	8 - RST
Pin 24 (BCM 8 / CE0)	7 - CE
Pin 16 (BCM 23)	6 - DC
Pin 19 (BCM 10 / MOSI)	5 - DIN
Pin 23 (BCM 11 / SCLK)	4 - CLK
Pin 1 (3.3 V)	3 - VCC
Pin 6 (GND)	2 - LIGHT
Pin 6 (GND)	1 - GND

This guide was written under Raspberry Pi OS Bookworm for the Raspberry Pi 4 and 5. It has not been checked with other/newer operating systems or hardware.

4.3 Code example Raspberry Pi

Installing the libraries To enable you to use the display as quickly and easily as possible, we are using the Adafruit_CircuitPython_PCD8544 library from Adafruit, which has been published under the MIT license. Open the console and execute the following commands:

```
sudo apt-get install python3-pip
sudo apt-get install fonts-dejavu
sudo apt-get install python3-pil
```

To activate the SPI interface, you must first go to raspi-config. To do this, enter the following command:

```
sudo raspi-config
```

Then select the option "3 Interface Options" and go to "I4 SPI" to activate the SPI interface. Confirm your selection by pressing "OK".

Then enter the following command in the terminal:

```
sudo nano /boot/firmware/config.txt
```

Now add the following line to the end of the file. Then save the file with **CTRL+O** and close with **CTRL+X** and **Enter**:

```
dtoverlay=spi0-0cs
```

Now restart your Raspberry Pi with the following command:

```
sudo reboot
```

Now you need to create a virtual environment for this project by entering the following commands:

```
mkdir dein_projekt

cd dein_projekt

python -m venv --system-site-packages env

source env/bin/activate
```

Now install the library used with the following command:

```
pip3 install adafruit-circuitpython-pcd8544
```

Using the sample codes

After completing the installation, you can create the sample file by entering the following command:

```
nano Displaytest.py
```

Now you have to copy the following code into the file. Of course you can modify this example according to your wishes and ideas. You can also download this code example here.

```
import time
import board
import busio
import digitalio

import adafruit_pcd8544

# SPI-Bus und Steuerpins initialisieren
spi = busio.SPI(board.SCK, MOSI=board.MOSI)
dc = digitalio.DigitalInOut(board.D23)
ce = digitalio.DigitalInOut(board.D8)
reset = digitalio.DigitalInOut(board.D24)

display = adafruit_pcd8544.PCD8544(spi, dc, ce, reset)

display.bias = 4
display.contrast = 60

# LED-Hintergrundbeleuchtung einschalten
# über einen extra GPIO Pin
backlight = digitalio.DigitalInOut(board.D26)
backlight.switch_to_output()
backlight.value = False

print("Pixel test")

# Löschen Sie die Anzeige.  Rufen Sie show immer nach der
Änderung von Pixeln auf
# um die Aktualisierung der Anzeige sichtbar zu machen!
display.fill(0)
display.show()

# Setzt ein Pixel an den Ursprung 0,0.
display.pixel(0, 0, 1)
# Setzt ein Pixel in die Mitte des Displays.
display.pixel(display.width // 2, display.height // 2, 1)
# Setzen eines Pixels in der gegenüberliegenden Eckposition.
display.pixel(display.width - 1, display.height - 1, 1)
display.show()
time.sleep(2)
```

```
print("Linien test")
# Wir zeichnen von Ecke zu Ecke, wobei wir hier ebenfalls
# alle Koordinatenpaare definieren
corners = (
    (0, 0),
    (0, display.height - 1),
    (display.width - 1, 0),
    (display.width - 1, display.height - 1),
)
display.fill(0)
for corner_from in corners:
    for corner_to in corners:
        display.line(corner_from[0], corner_from[1], corner_to[0], corner_to[1], 1)
display.show()
time.sleep(2)

print("Rechtecks test")
display.fill(0)
w_delta = display.width / 10
h_delta = display.height / 10
for i in range(11):
    display.rect(0, 0, int(w_delta * i), int(h_delta * i), 1)
display.show()
time.sleep(2)

print("Text test")
display.fill(0)
display.text("Presenting an ", 0, 0, 1)
display.text("  Examplecode ", 0, 8, 1)
display.text("for the SBC-  ", 0, 16, 1)
display.text("LCD84x48 based", 0, 24, 1)
display.text("on the pcd8455", 0, 32, 1)
display.text("Nokia 5110 LCD", 0, 40, 1)
display.show()

while True:
    display.invert = True
    time.sleep(0.5)
    display.invert = False
    time.sleep(0.5)
```

Once you have copied the code into your file, you can save it by pressing "**CTRL+O**" and close it by pressing "**CTRL+X**". To ensure that the code example works properly, you must then download and unzip the font file. To do this, execute the following two commands. **To ensure that everything works, the unzipped font file must be placed in the same folder as the code to be executed.**

```
wget https://joy-it.net/files/files/Produkte/SBC-LCD84x48/font5x8.zip
unzip font5x8.zip
```

The following command then executes the file with your code and you can end the code with **CTRL+C**.

```
python3 Displaytest.py
```

7. ADDITIONAL INFORMATION

Our information and take-back obligations according to the Electrical and Electronic Equipment Act (ElektroG)

Symbol on electrical and electronic equipment:

This crossed-out dustbin means that electrical and electronic appliances do not belong in the household waste. You must return the old appliances to a collection point.

Before handing over waste batteries and accumulators that are not enclosed by waste equipment must be separated from it.

Return options:

As an end user, you can return your old device (which essentially fulfills the same function as the new device purchased from us) free of charge for disposal when you purchase a new device.

Small appliances with no external dimensions greater than 25 cm can be disposed of in normal household quantities independently of the purchase of a new appliance.

Possibility of return at our company location during opening hours:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn, Germany

Possibility of return in your area:

We will send you a parcel stamp with which you can return the device to us free of charge. Please contact us by email at Service@joy-it.net or by telephone.

Information on packaging:

If you do not have suitable packaging material or do not wish to use your own, please contact us and we will send you suitable packaging.

8. SUPPORT

If there are still any issues pending or problems arising after your purchase, we will support you by e-mail, telephone and with our ticket support system.

Email: **service@joy-it.net**
Ticket system: **http://support.joy-it.net**
Telephone: +49 (0)2845 9360-50 (Mon - Thur: 09:00 - 17:00 o‘clock,
Fri: 09:00 - 14:30 o‘clock)

For further information please visit our website:
www.joy-it.net

Published: 3.18.2024