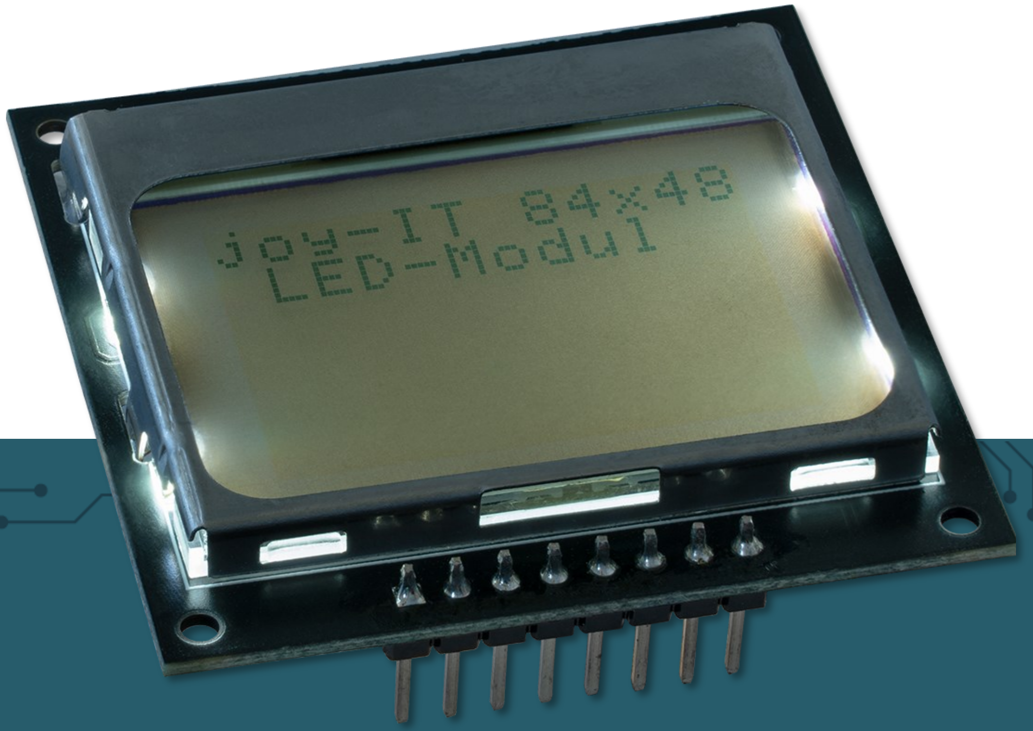


84X48 LCD DISPLAY

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.

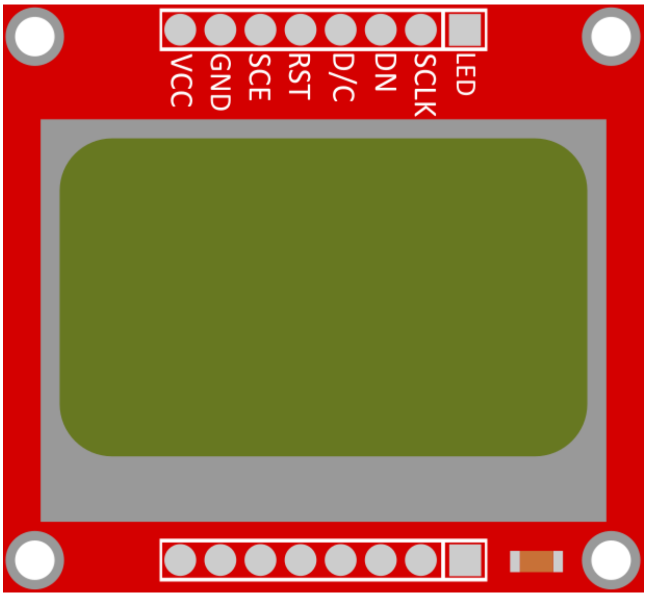
The display can be used with many common developer boards such as Raspberry Pi or Arduino, for example, to display measured values or measured value graphics directly on the microcontroller or single-board computer.

2. DIFFERENCES BETWEEN THE REVISIONS

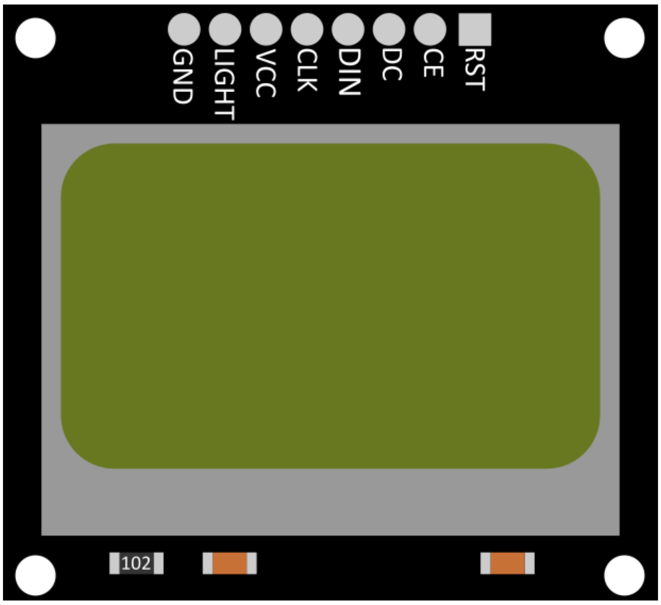
This product has a new revision that replaces the previous revision. These changes include the pin out, the LED and the silkscreen.

Below you can see 2 drawings of the different revisions.

Revision 1



Revision 2

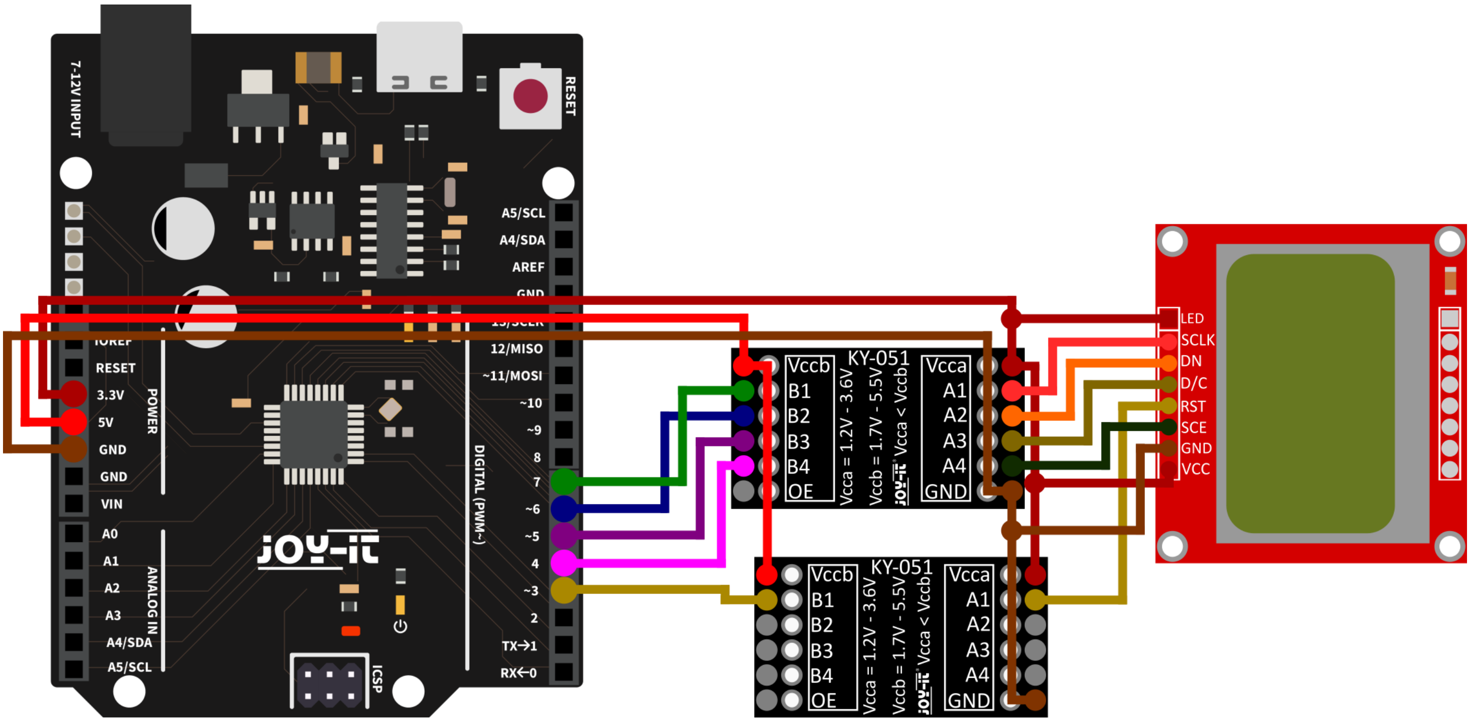


3. USE WITH AN ARDUINO

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.



KY-051	84 x 48 LCD
A1	7 - SCLK
A2	6 - DN
A3	5 - D/C
A4	3 - SCE

second KY-051	84 x 48 LCD
A1	4 - RST

Arduino	84 x 48 LCD
5V	8 - LED
GND	2 - GND
5V	1 - VCC

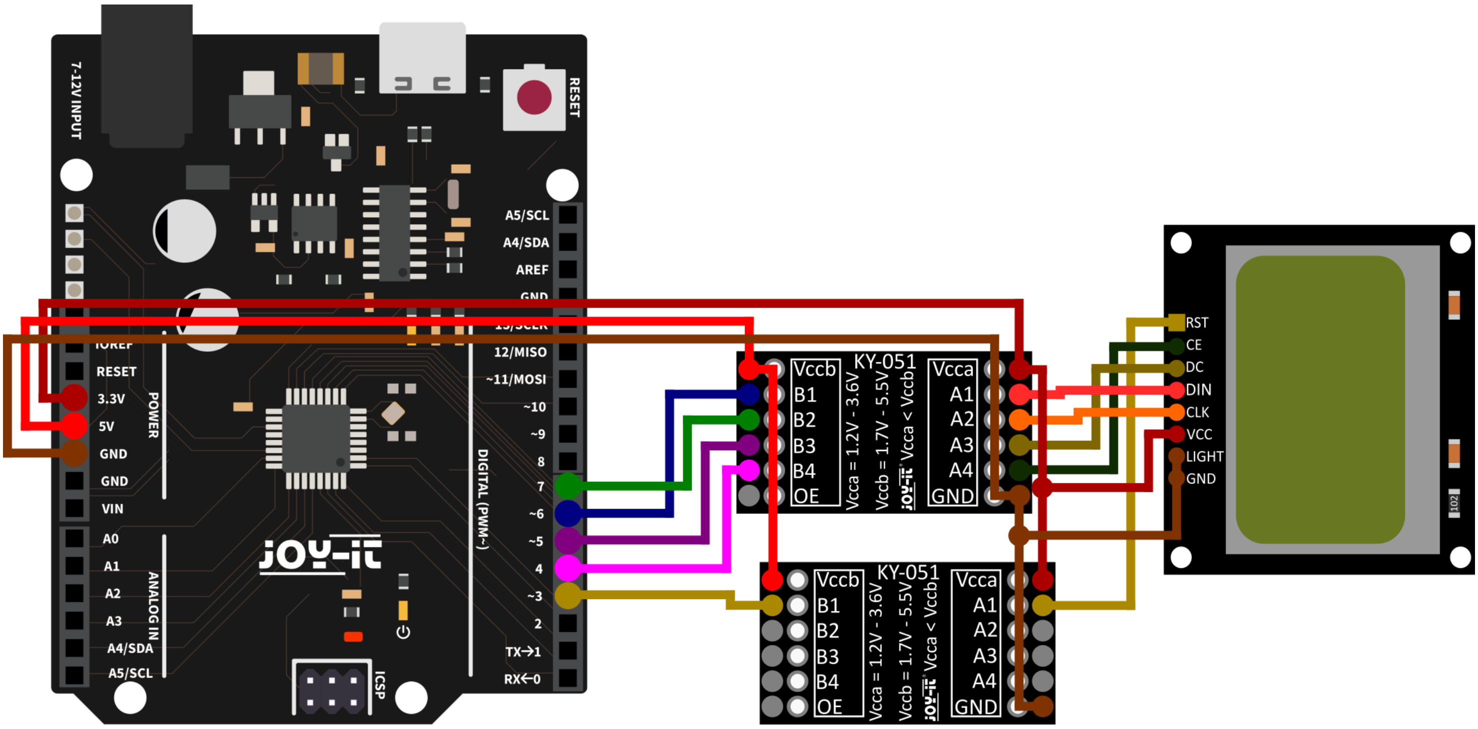
Arduino	KY-051
GND	GND
3,3V	Vcca
5V	Vccb
Digital Pin 7	B1
Digital Pin 6	B2
Digital Pin 5	B3
Digital Pin 4	B4

Arduino	second KY-051
GND	GND
3,3V	Vcca
5V	Vccb
Digital Pin 3	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.



KY-051	84 x 48 LCD
A1	5 - DIN
A2	4 - CLK
A3	6 - D/C
A4	7 - CE

second KY-051	84 x 48 LCD
A1	8 - RST

Arduino	84 x 48 LCD
5V	3 - VCC
GND	2 - LIGHT
GND	1 - GND

Arduino	KY-051
GND	GND
3,3V	Vcca
5V	Vccb
Digital Pin 6	B1
Digital Pin 7	B2
Digital Pin 5	B3
Digital Pin 4	B4

Arduino	second KY-051
GND	GND
3,3V	Vcca
5V	Vccb
Digital Pin 3	B1

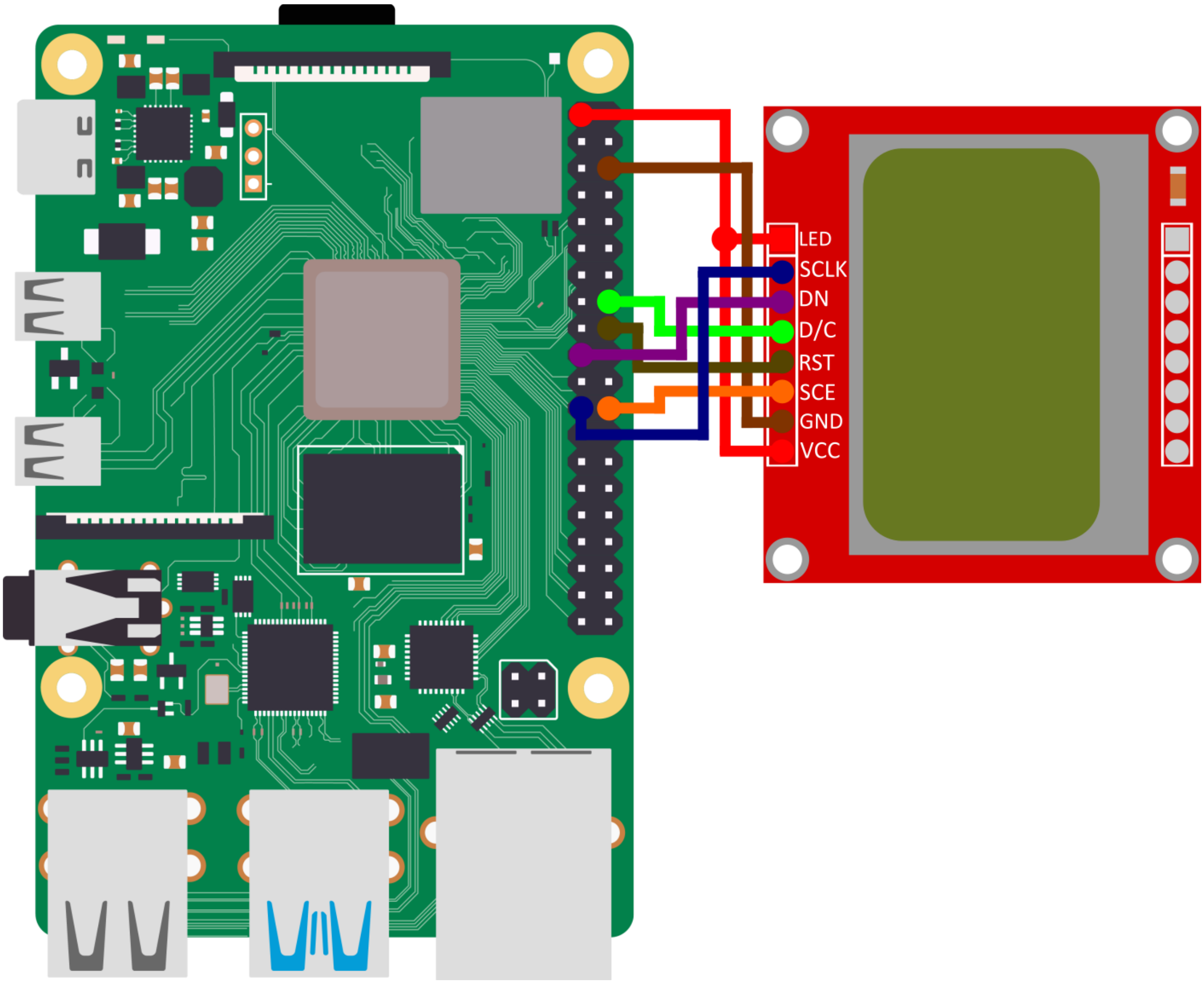
3.3 Code example Arduino

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)

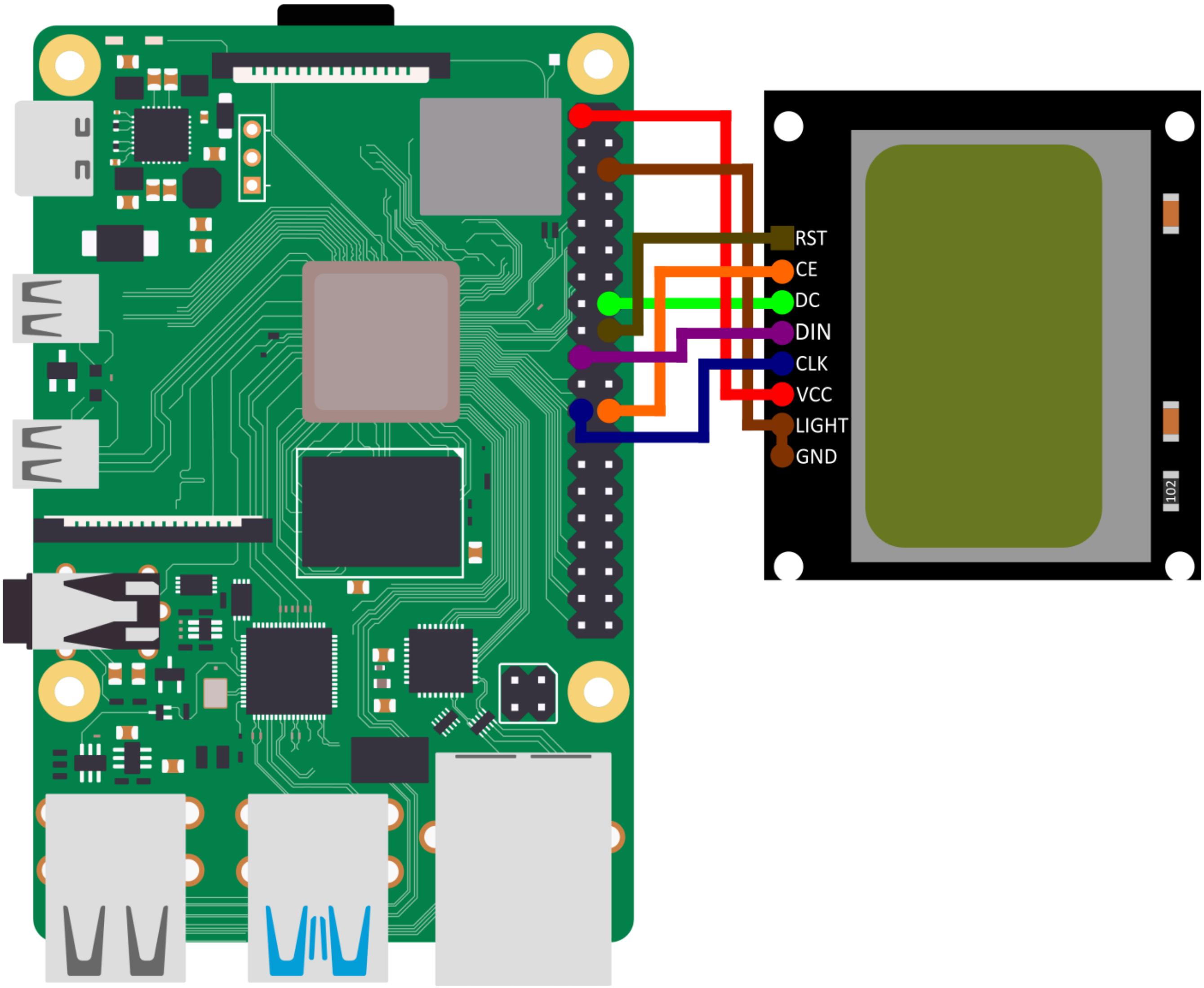
Connect the display to the pins of your Raspberry Pi as shown in the following picture or table.



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

4.3 Code example Raspberry Pi

Installation of the libraries

To allow you to use the display as quickly and easily as possible, we will resort to the **Adafruit_CircuitPython_PCD8544** library from **Adafruit**, released under the **MIT license**. Open the console and execute the following commands:

```
sudo apt update
sudo apt upgrade -y
sudo usermod -a -G i2c,spi,gpio pi
sudo apt install python3-dev python3-pip libfreetype6-dev libjpeg-dev build-essential -y
sudo apt install git -y
sudo pip3 install adafruit-circuitpython-pcd8544
sudo raspi-config
```

To enable the SPI and I2C interface, you must first enter the raspi-config. Then select the option "**3 Interface Options**" and go to "**I4 SPI**" to enable the SPI interface. Confirm your selection by pressing "**OK**". To enable the I2C interface, go to "**3 Interface Options**" again and select "**I5 I2C**" and confirm your selection by pressing "**OK**" as well.
After you have enabled both I2C and SPI, you can restart your Raspberry Pi by entering the following command.

```
sudo reboot
```

Using the sample codes

Now the installation of the display is complete. Now open the console again and create a new folder and file with the following commands:

```
mkdir SBC-LCD84x48
cd SBC-LCD84x48
nano Displaytest.py
```

Now you still need 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**.

```python
import time
import board
import busio
import digitalio

import adafruit_pcd8544

# Initialize SPI bus and control pins
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

# Switch on LED backlight via an extra GPIO pin
backlight = digitalio.DigitalInOut(board.D26)
backlight.switch_to_output()
backlight.value = False

print("Pixel test")
```

```
# Delete the display.  Always call show after changing pixels
# to make the display update visible!
display.fill(0)
display.show()

# Sets a pixel to the origin 0,0.
display.pixel(0, 0, 1)
# Places a pixel in the center of the display.
display.pixel(display.width // 2, display.height // 2, 1)
# Set a pixel in the opposite corner position.
display.pixel(display.width - 1, display.height - 1, 1)
display.show()
time.sleep(2)

print("Lines test")
# We draw from corner to corner, also defining here all pairs of coordinates
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("Rectangle 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)
```

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

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

The following command then executes the file with its code

```
python3 Displaytest.py
```

and you can end the code with **CTRL+C**.

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 (10-17 o‘clock)

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

Published: 4.24.2023