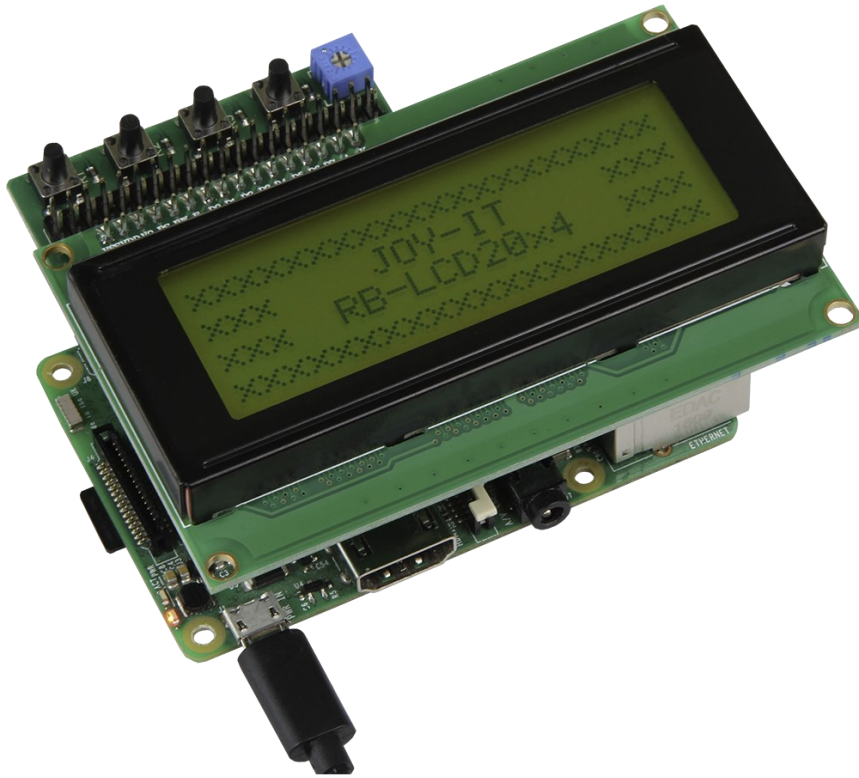# LCD DISPLAY 20X4
## with Buttons



## 1. GENERAL INFORMATION

Dear customer,

Thank you for purchasing our product. In the following, we will show you which things should be noted during the use.

Should you encounter any unexpected problems, do not hesitate to contact us.

This manual is designed for version 2 of this display (PCF8574).

## 2. CONNECTING THE DISPLAY

Put the display on the pin header that way, that the display is right above your Raspberry Pi. As soon as the Raspbbery Pi is connected to a power supply, the backgroundlight of the display will light up.
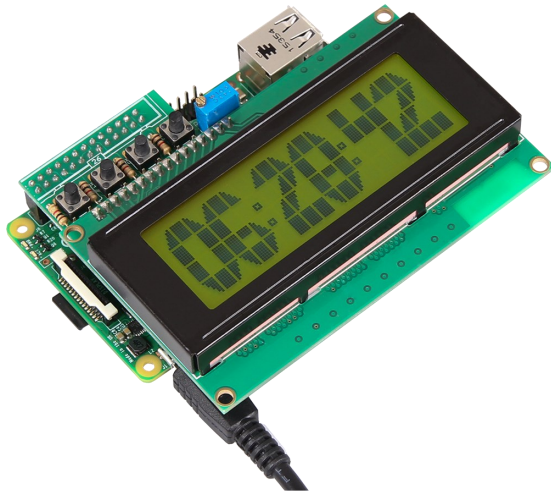


## 3. INSTALLATION OF THE SYSTEM

If your Raspberry Pi has the latest Raspbian software, you can skip this step. Otherwise, please follow the instructions.

Install card the latest Raspbian system image on your SD. You can download it [here](#).
With an appropriate program (e.g. Etcher), you can transfer the image onto the SD card. After a successful transfer, the SD card can be put into the Raspberry Pi and the system can start.
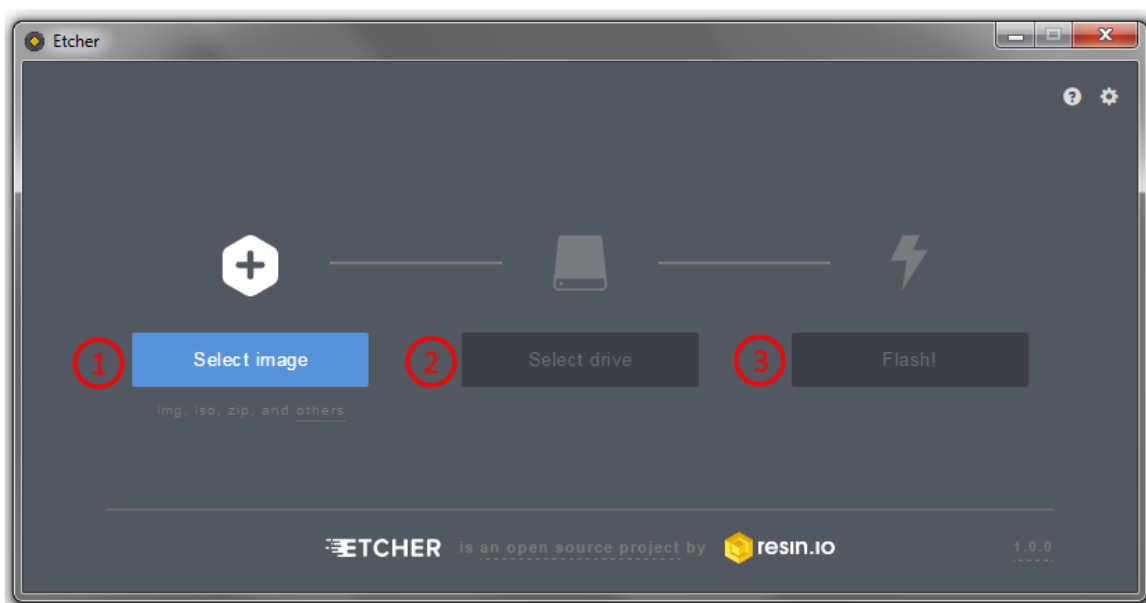
## 4. USE AND CODE EXAMPLE - VERSION 2

To use the display completely, you must install first two libraries for the display.  First, you need the **Python GPIO** library from Adafruit which is published under the **MIT license**. You install this library with the following commands:

```
git clone https://github.com/adafruit/Adafruit_Python_GPIO
cd Adafruit_Python_GPIO
sudo python3 setup.py install
```

You can leave the subfolder with the following command:

```
cd ..
```

The second library is the **Python CharLCD** library from Adafruit which is again published under the **MIT license**. You can install it like this:

```
git clone https://github.com/adafruit/Adafruit_Python_CharLCD
cd Adafruit_Python_CharLCD/
sudo python3 setup.py install
```

You leave the folder again with:

```
cd ..
```

Before the use of the display, you have to enable I2C. Therefore, enter the following command to open the Raspberry Pi configuration settings:
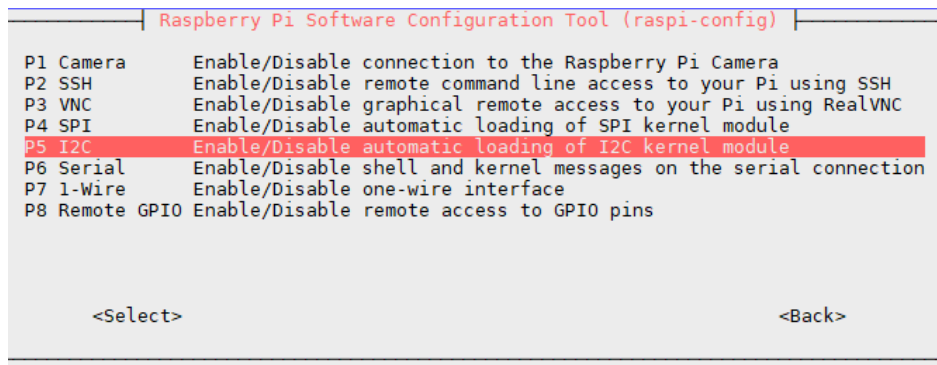
```
sudo raspi-config
```

Select  the option *Interfacing Options* and confirm with **Enter**:

```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├
 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your location
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool


          <Select>                                   <Finish>
```

After that, select the option *I2C*:

```
─────────┤ Raspberry Pi Software Configuration Tool (raspi-config) ├─────────

  P1 Camera       Enable/Disable connection to the Raspberry Pi Camera
  P2 SSH          Enable/Disable remote command line access to your Pi using SSH
  P3 VNC          Enable/Disable graphical remote access to your Pi using RealVNC
  P4 SPI          Enable/Disable automatic loading of SPI kernel module
  P5 I2C          Enable/Disable automatic loading of I2C kernel module
  P6 Serial       Enable/Disable shell and kernel messages on the serial connection
  P7 1-Wire       Enable/Disable one-wire interface
  P8 Remote GPIO  Enable/Disable remote access to GPIO pins




            <Select>                                          <Back>

```

The next message has to be confirmed with *Yes* and after that, you can leave the settings with *Finish*. I2C is now succesfully activated and you can use now the display.

```
  Would you like the ARM I2C interface to be enabled?





              <Ja>                   <Nein>
```
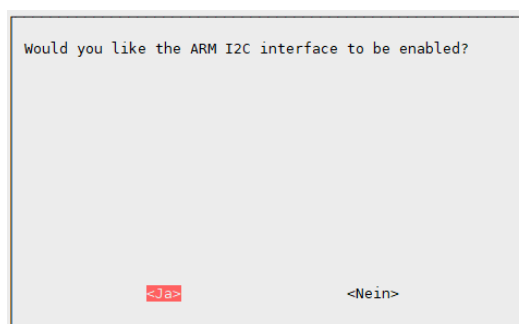
It could be necessary that you have to adjust the contrast manually before you can see the output of the display.
Adjust the contrast therefore, that you can rotate the marked potentiometer with a small screwdriver, like it is shown in the picture below.
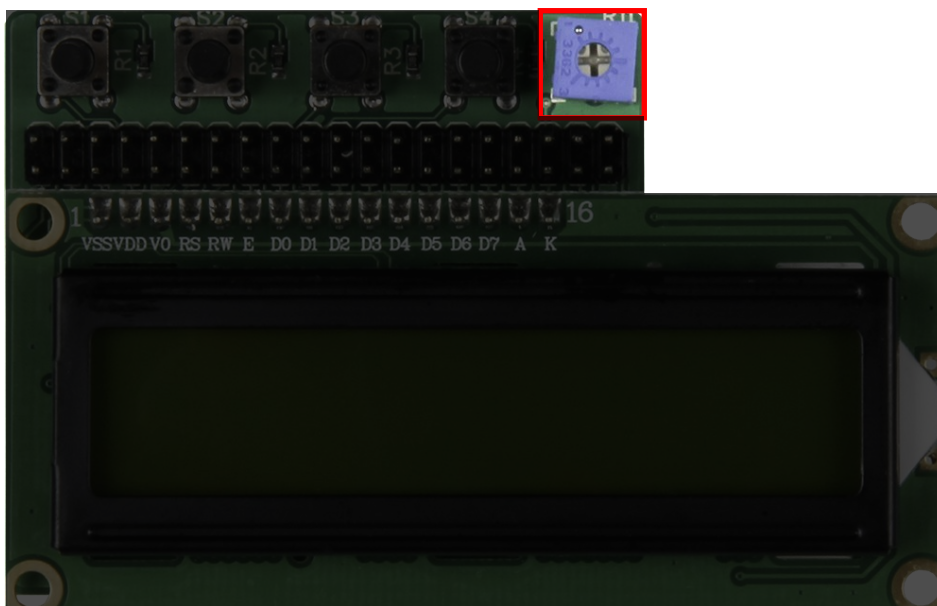
To use the display completely, you can download the example code here or you can create yourself a file and copy the following code into that file.
To create a new file, you use the following command:

```
sudo nano lcd20x4.py
```

```python
from Adafruit_CharLCD import Adafruit_CharLCD
import Adafruit_GPIO.PCF8574 as PCF
import RPi.GPIO as GPIO

LCD = PCF.PCF8574(address=0x27)
LCD.setup(5,0)
LCD.output(5,0)

# Declares which PCF pin is connected to which LCD pin
lcd_rs        = 4      # RegisterSelect pin is connected to GPIO4
lcd_en        = 7      # Enable pin is connected to GPIO7
d4,d5,d6,d7   = 0,1,2,3   # Data pins 4,5,6,7 is connected to GPIO 0,1,2,3
cols,lines    = 20,4 # Number of columns and rows of the display

# Declares the pins of the Raspberry Pi to the buttons
SW1 = 4       #GPIO  4
SW2 = 16      #GPIO 16
SW3 = 10      #GPIO 10
SW4 = 9       #GPIO  9
GPIO.setmode(GPIO.BCM)
GPIO.setup(SW1, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(SW2, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(SW3, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(SW4, GPIO.IN, pull_up_down = GPIO.PUD_UP)

# Initializes the LCD
lcd = Adafruit_CharLCD(lcd_rs, lcd_en, d4, d5, d6, d7,cols, lines, gpio=LCD)
lcd.clear()
lcd.message('xx RB-LCDV2 Test xx\n 1234567890\n abcdefgh \n 0987654321')
# Watch the buttons:
try:
  while True:
    if GPIO.input(SW1) == GPIO.LOW:
      lcd.clear()
      lcd.message ("Button 1 pressed!")
    if GPIO.input(SW2) == GPIO.LOW:
      lcd.clear()
      lcd.message ("Button 2 pressed!")
    if GPIO.input(SW3) == GPIO.LOW:
      lcd.clear()
      lcd.message ("Button 3 pressed!")
    if GPIO.input(SW4) == GPIO.LOW:
      lcd.clear()
      lcd.message ("Button 4 pressed!")
except KeyboardInterrupt:
  lcd.clear()
  GPIO.cleanup()
```

The file can be saved with the key combination **CTRL + O** and you can leave the editor with **CTRL + X**.
After that, you can enter the following command:

```
sudo python3 lcd20x4.py
```

## 5. OTHER INFORMATION

### Our Information and Take-back Obligations according to the Electrical and Electronic Equipment Act (ElektroG)

**Symbol on Electrial and Electronic Products:**
This crossed-out bin means that electrical and electronic products do not belong into the household waste. You must hand over your oldappliance to a registration place. Before you can hand over the old appliance, you must remove used batteries and replacement batteries which are not enclosed by the device.

**Return Options:**
As the end user, you can hand over your old appliance (which has essentially the same functions as the new one bought with us) free of charge for disposal with the purchase of a new device.
Small devices, which do not have outer dimensions bigger than 25 cm can be handed in for disposal independently of the
purchase of a new product in normal household quantities.

**1. Possibility of return at our company location during our opening hours**
SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

**2. Possibility of return nearby**
We will send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, please contact us via e-mail at service@joy-it.net or via telephone.

**Information about Package:**
Please package your old appliance safe for transport. Should you not have suitable packaging material or you do not want to use your own material, you can contact us and we will send you an appropriate package.

## 6. SUPPORT

If any questions remained open or problems may arise after your purchase,we are available by email, telephone and ticket support system to answer these.

Email: service@joy-it.net
Ticket-system: http://support.joy-it.net
Telephone: +49 (0)2845 98469 – 66 (10 - 17 o'clock)

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