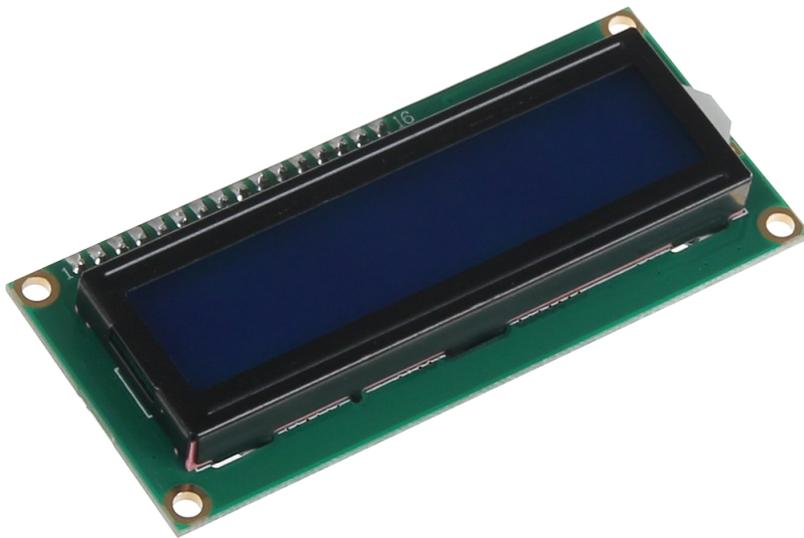


16X2 LCD MODUL

16x2 LCD Modul mit 16 Pin Stiftleiste



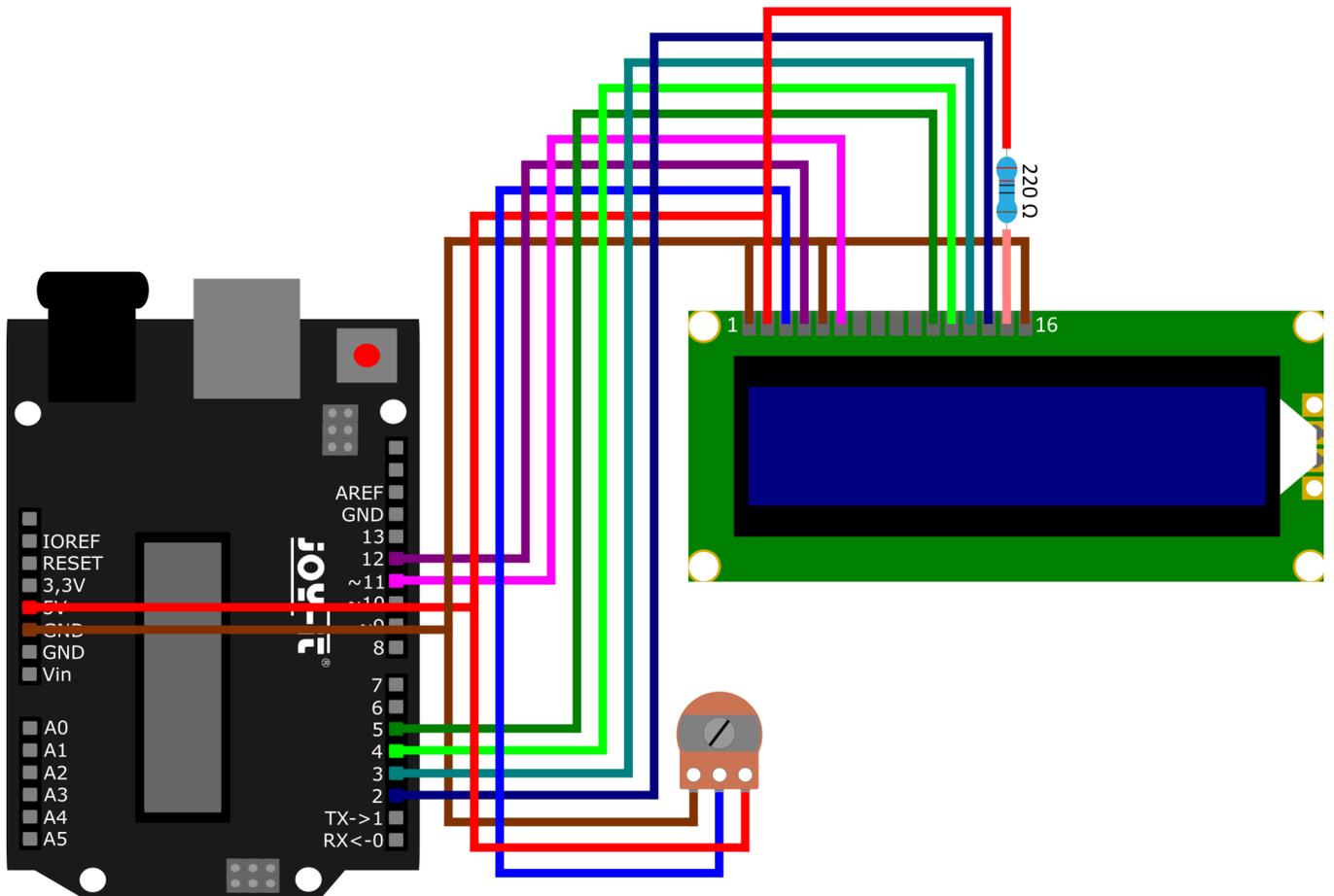
1. ALLGEMEINE INFORMATIONEN

Sehr geehrte *r Kunde *in,
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

2. VERWENDUNG MIT DEM ARDUINO

2.1. Anschluss



Arduino	Display
D12	Pin 4
D11	Pin 6
D5	Pin 11
D4	Pin 12
D3	Pin 13
D2	Pin 14
GND	Pin 5
GND	Pin 1
5V	Pin 2
5V über einen 220Ω Widerstand	Pin 15
GND	Pin 16

Arduino	Potentiometer	Display
GND	-	
	Signal	Pin 3
5V	+	

2.2. Codebeispiel

Für die Verwendung können Sie die Bibliothek [LiquidCrystal](#) von [Arduino Libraries](#) verwenden, welche unter der [Creative Commons Attribution-ShareAlike 3.0 License](#) veröffentlicht wurde.

Sie können sich die Bibliothek [hier](#) herunterladen. Sie können die heruntergeladene Bibliothek in Ihrer Arduino IDE einbinden, indem Sie die .zip-Datei unter **Sketch** → **Bibliothek einbinden** → **.ZIP-Bibliothek hinzufügen...** Wählen Sie dort also die heruntergeladene Bibliothek aus und starten Sie Ihre Arduino IDE neu, um die Bibliothek nutzen zu können. Sie können auch unter **Werkzeuge** → **Bibliotheken verwalten...** nach LiquidCrystal suchen und die Bibliothek so einbinden.

Kopieren Sie sich den folgenden Code in Ihre IDE, um Ihr Display zu testen. Um das Codebeispiel auszuführen, klicken Sie auf **Hochladen**. Achten Sie dabei darauf, dass unter **Werkzeuge Board** und **Port** richtig ausgewählt sind.

```
// Importieren der benötigten Bibliothek
#include <LiquidCrystal.h>

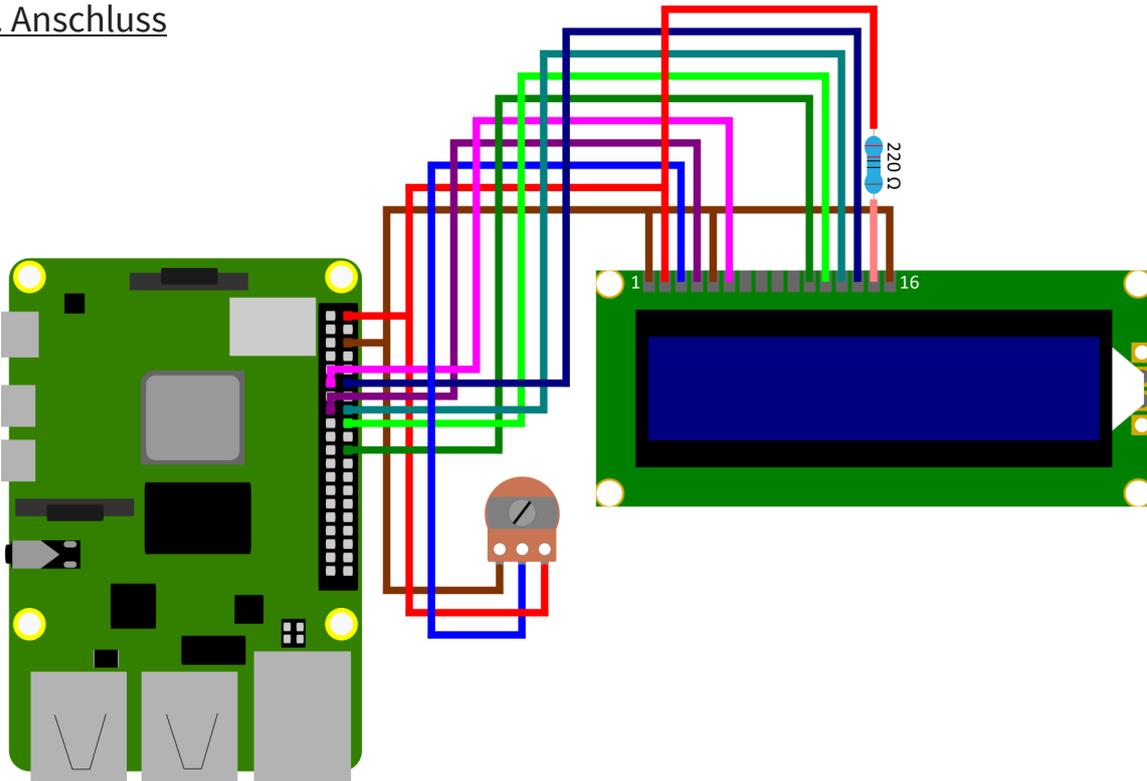
// die Bibliothek initialisieren, indem jeder benötigte LCD-
// Schnittstellen-Pin
// mit der Arduino-Pin-Nummer, an die er angeschlossen ist, ver-
// knüpft wird
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // die Anzahl der Spalten und Zeilen des LCDs festlegen
  lcd.begin(16, 2);
}

void loop() {
  // Cursor auf Spalte 7, Zeile 1 setzen
  lcd.setCursor(7, 0);
  // Ausgabe des Textes auf dem LCD
  lcd.print("joy-IT");
  lcd.setCursor(8, 1);
  // Ausgabe der Zeit in Sekunden seid dem Letzten Neustart
  lcd.print(millis() / 1000);
}
```

3. VERWENDUNG MIT DEM RASPBERRY PI

3.1. Anschluss



ACHTUNG!!!

Da das Display mit einem 5V Logiklevel arbeitet, müssen Sie wenn Sie mit dem Raspberry Pi von dem Display Daten lesen wollen einen Logiklevelwandler verwenden.

In diesem Beispiel wird das Display jedoch nur beschrieben, deshalb kann der Raspberry Pi direkt an das Display angeschlossen werden.

Raspberry Pi	Display
GPIO 22 (Pin 15)	Pin 4
GPIO 17 (Pin 13)	Pin 6
GPIO 25 (Pin 22)	Pin 11
GPIO 24 (Pin 18)	Pin 12
GPIO 23 (Pin 16)	Pin 13
GPIO 18 (Pin 12)	Pin 14
GND	Pin 5
GND	Pin 1
5V	Pin 2
5V durch einen 220Ω Widerstand	Pin 15
GND	Pin 16

Raspberry Pi	Potentiometer	Display
GND	-	
	Signal	Pin 3
5V	+	

3.2. Codebeispiel

Wir benutzen hier die Bibliotheken [adafruit-blinka](#) und [adafruit-circuitpython-charlcd](#) welche von [adafruit](#) veröffentlicht wurden. Sie wurden unter der [MIT Lizenz](#) veröffentlicht.

Doch bevor diese benutzt werden kann, müssen Sie zuerst folgende Befehle in der Konsole auf ihrem Raspberry Pi ausführen.

```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install python3-pip
```

Nachdem Sie dies getan haben können Sie nun `adafruit-blinka` und `adafruit-circuitpython-charlcd` auf ihrem Raspberry Pi installieren indem Sie folgende Befehle ausführen.

```
sudo pip3 install adafruit-blinka
sudo pip3 install adafruit-circuitpython-charlcd
```

Nun können Sie, nachdem Sie die beiden Bibliotheken installiert haben, eine neue Datei anlegen, indem Sie folgenden Befehl benutzen

```
nano com-LCD1602.py
```

und dort können Sie dann einfach den folgenden Code herein kopieren.

```
#!/usr/bin/env python3
#-*- coding:utf-8 -*-
from subprocess import Popen, PIPE
from time import sleep
from datetime import datetime
import RPi.GPIO as GPIO
import board
import digitalio
import adafruit_character_lcd.character_lcd as characterlcd

# Ändern Sie dies, wenn Sie ein LCD in einer anderen Größe haben.
lcd_columns = 16
lcd_rows = 2

lcd_rs = digitalio.DigitalInOut(board.D22)
lcd_en = digitalio.DigitalInOut(board.D17)
lcd_d4 = digitalio.DigitalInOut(board.D25)
lcd_d5 = digitalio.DigitalInOut(board.D24)
lcd_d6 = digitalio.DigitalInOut(board.D23)
lcd_d7 = digitalio.DigitalInOut(board.D18)

# Initialisieren des LCDs
lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
lcd_d7, lcd_columns, lcd_rows)

# Suche nach einem aktiven Ethernet- oder WiFi-Gerät
def find_interface():
    find_device = "ip addr show"
    interface_parse = run_cmd(find_device)
    for line in interface_parse.splitlines():
```

```

        if "state UP" in line:
            dev_name = line.split(':')[1]
    return dev_name

# Finden einer Aktiven IP-Adresse auf dem aktuellen Gerät
def parse_ip():
    find_ip = "ip addr show %s" % interface
    find_ip = "ip addr show %s" % interface
    ip_parse = run_cmd(find_ip)
    for line in ip_parse.splitlines():
        if "inet " in line:
            ip = line.split(' ')[5]
            ip = ip.split('/')[0]
    return ip

# Unix-Shell-Befehl ausführen, Rückgabe als ASCII
def run_cmd(cmd):
    p = Popen(cmd, shell=True, stdout=PIPE)
    output = p.communicate()[0]
    return output.decode('ascii')

# LCD-Bildschirm Leeren, bevor wir beginnen
lcd.clear()

# bevor wir die Hauptschleife starten - aktives Netzwerkgerät und IP-Adresse
erkennen
sleep(2)
interface = find_interface()
ip_address = parse_ip()
count = 0

if __name__ == '__main__':
    try:
        while True:
            if count == 0:
                lcd.cursor_position(5, 0)
                lcd.message = "joy-IT"

                lcd.cursor_position(3, 1)
                lcd.message = "for makers"
                sleep(3)
                lcd.cursor_position(1, 1)
                lcd.message = "& Professionals"
                sleep(3)
                lcd.clear()
                count = 1
                sleep(2)

            lcd.cursor_position(0, 0)
            # Ausgabe des Datums und der Uhrzeit auf dem LCD
            lcd.message = datetime.now().strftime('%b %d %H:%M:%S\n')

            lcd.cursor_position(0, 1)
            # Ausgabe der aktuellen IP-Adresse auf dem LCD
            lcd.message = "IP " + ip_address

            sleep(1)

    except KeyboardInterrupt:
        lcd.clear()
        GPIO.cleanup()

```

Nachdem Sie dann den Code in Ihre Datei kopiert haben, können Sie mit **STRG+O** die Datei speichern und mit **STRG+X** die Datei schließen. Danach können Sie die Datei mit dem Befehl

```
python3 com-LCD1602.py
```

ausführen und auf Ihrem LCD Display anschauen.

Um den Code zu unterbrechen, drücken Sie einfach **STRG+C**.

Alternativ können Sie das Programm auch mit diesen 3 einfachen Befehlen auf ihren Raspberry Pi herunterladen, entpacken und dann anschließend starten.

```
wget https://joy-it.net/files/files/Produkte/com-LCD16x2/com-LCD1602-DE.zip
```

```
unzip com-LCD1602-DE.zip
```

```
python3 com-LCD1602-DE.py
```

4. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)



Symbol auf Elektro- und Elektronikgeräten:

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

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

Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

5. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 9360-50 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net