



LCD 128X64

Ansteuerbar durch SPI und parallel

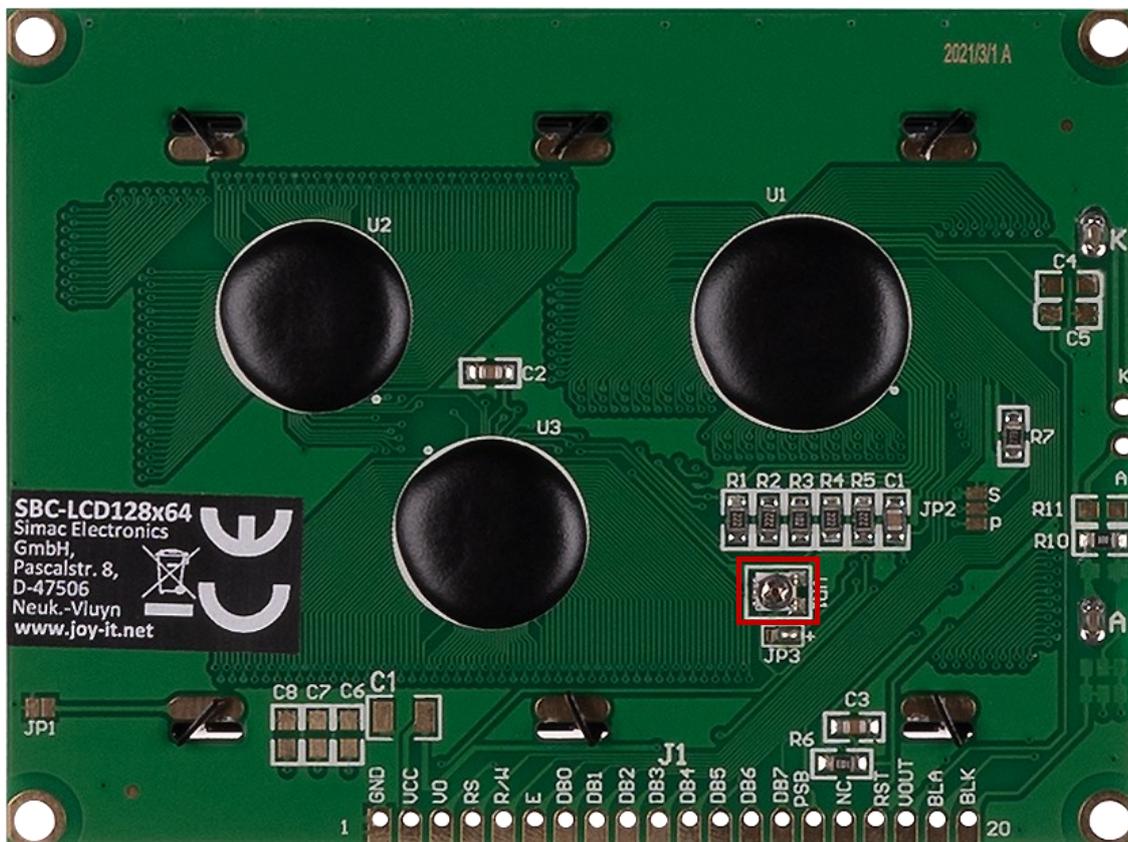
1. ALLGEMEINE INFORMATIONEN

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. KONTRASTEINSTELLUNGEN

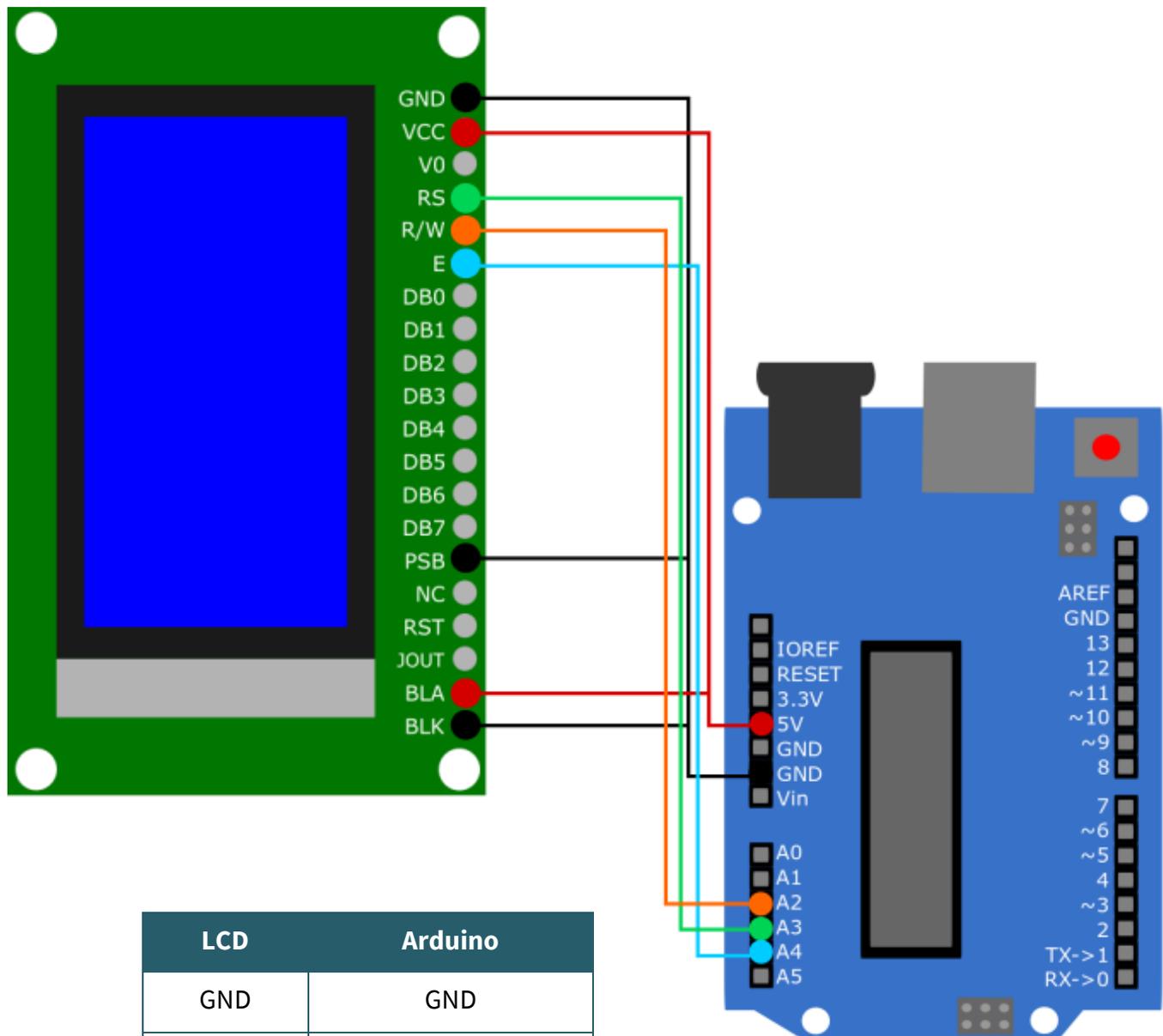
Auf der Rückseite des Displays befindet sich ein Potentiometer mit dem Sie den Kontrast einstellen können.



3. VERWENDUNG MIT DEM ARDUINO

3.1 SPI

3.1.1 Anschluss



LCD	Arduino
GND	GND
VCC	5V
RS	A3
R/W	A2
E	A4
PSB	GND
BLA	5V
BLK	GND

3.1.2 Bibliothek

Wir verwenden mit dem Arduino die **U8g2-Library**, welche von [oliver](#) veröffentlicht wurde. Diese Bibliothek können Sie in Ihrer Arduino IDE unter **Werkzeuge** → **Bibliotheken verwalten...** finden. Wenn Sie die Bibliothek erfolgreich heruntergeladen haben, können Sie unter **Datei** → **Beispiele** → **U8g2** Beispiele finden.

3.1.3 Code Beispiel

Hier wird Ihnen nun erklärt, wie Sie exemplarisch ein Codebeispiel der U8g2 Bibliothek verwenden müssen. Wir verwenden das Beispiel *GraphicsTest*.

Dieses Beispiel finden Sie unter **Datei** → **Beispiele** → **U8g2** → **full_buffer** → **GraphicsTest**. In diesem Code müssen Sie die Kommentarstriche in Zeile 185 entfernen, bevor Sie diese mittels der Taste **Hochladen** ausführen können.

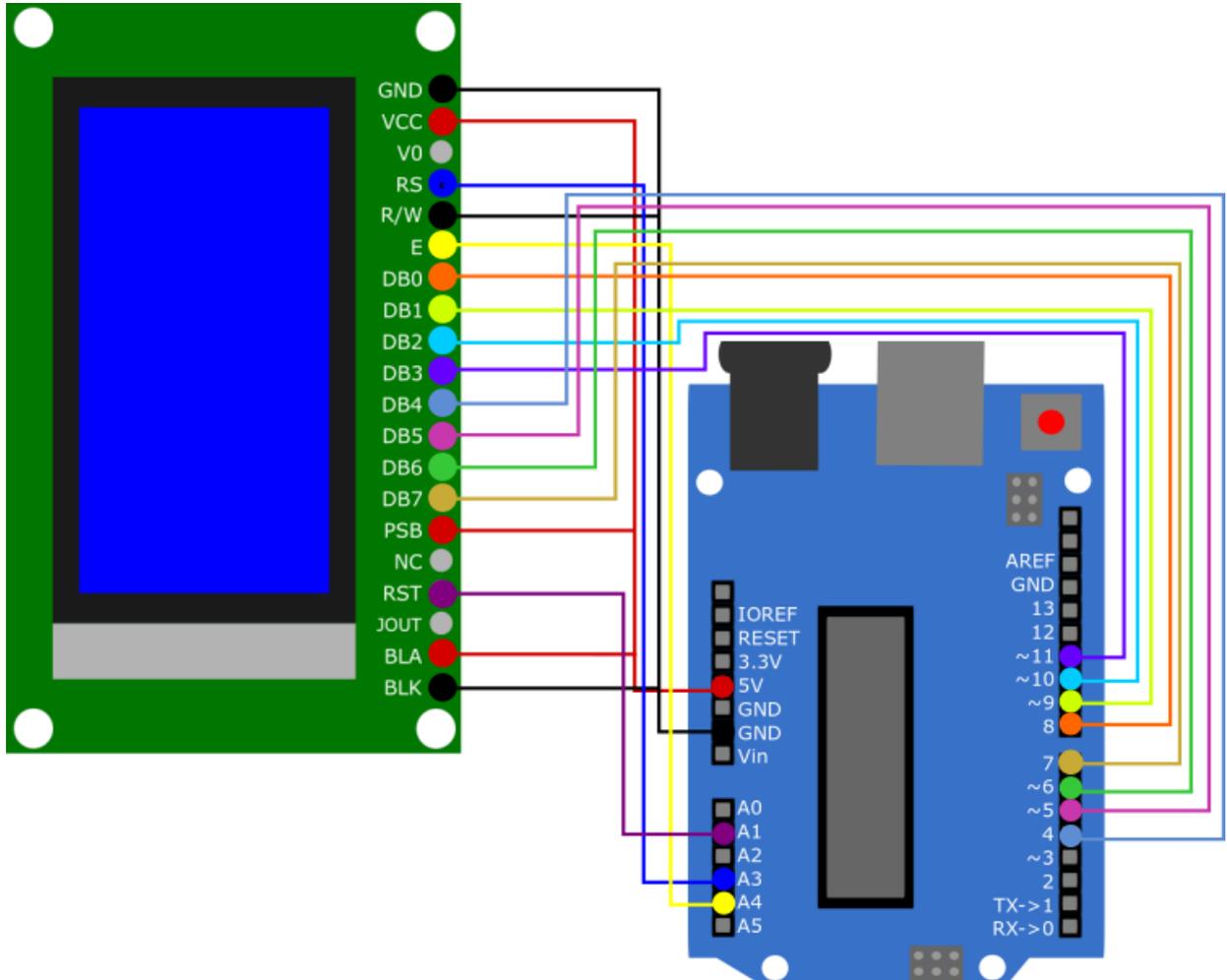
Die Zeile 185 enthält nämlich in diesem Falle die Deklaration des Displays mit SPI. Sie können die anderen Beispiele natürlich genauso verwenden, indem Sie in derselben Zeile in den anderen Codebeispielen auch die Kommentarstriche entfernen.

```
181 //U8G2_5T7920_192X32_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 18 /* A4 */, /* data=*/ 16 /* A2 */, /* CS=*/ 17 /* A3 */, /* reset=*/ U8XB_PIN_NONE);
182 //U8G2_5T7920_160X32_F_8080 u8g2(U8G2_R0, 8, 9, 10, 11, 4, 5, 6, 7, /*enable=*/ 18, /*cs=*/ U8XB_PIN_NONE, /*dc=*/ 17, /*reset=*/ U8XB_PIN_NONE);
183 //U8G2_5T7920_160X32_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 18 /* A4 */, /* data=*/ 16 /* A2 */, /* CS=*/ 17 /* A3 */, /* reset=*/ U8XB_PIN_NONE);
184 //U8G2_5T7920_128X64_F_8080 u8g2(U8G2_R0, 8, 9, 10, 11, 4, 5, 6, 7, /*enable=*/ 18 /* A4 */, /*cs=*/ U8XB_PIN_NONE, /*dc/rs=*/ 17 /* A3 */, /*reset=*/ 15 /* A1 */); // Remember to set R/W to 0
185 U8G2_5T7920_128X64_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 18 /* A4 */, /* data=*/ 16 /* A2 */, /* CS=*/ 17 /* A3 */, /* reset=*/ U8XB_PIN_NONE);
186 //U8G2_5T7920_128X64_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 13, /* data=*/ 11, /* CS=*/ 10, /* reset=*/ 8);
187 //U8G2_5T7920_128X64_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 14, /* data=*/ 13, /* CS=*/ 15, /* reset=*/ 16); // Feather HUZZAH ESP8266, E=clock=14, RW=data=13, RS=CS
188 //U8G2_5T7920_128X64_F_HW_SPI u8g2(U8G2_R0, /* CS=*/ 10, /* reset=*/ 8);
```

Beachten Sie, dass unter **Werkzeuge Board** und **Port** richtig ausgewählt sein müssen.

3.2 Parallel (8080)

3.2.1 Anschluss



LCD	Arduino
GND	GND
VCC	5V
RS	A3
R/W	GND
E	A4
DB0	Pin 8
DB1	Pin 9
DB2	Pin 10
DB3	Pin 11
DB4	Pin 4
DB5	Pin 5
DB6	Pin 6
DB7	Pin 7
PSB	5V
BLA	5V
BLK	GND

3.1.2 Bibliothek

Wir verwenden mit dem Arduino die [U8g2-Library](#), welche von [oliver](#) veröffentlicht wurde. Diese Bibliothek können Sie in Ihrer Arduino IDE unter **Werkzeuge** → **Bibliotheken verwalten...** finden. Wenn Sie die Bibliothek erfolgreich heruntergeladen haben, können Sie unter **Datei** → **Beispiele** → **U8g2** Beispiele finden.

3.1.3 Code Beispiel

Hier wird Ihnen nun erklärt, wie Sie exemplarisch ein Codebeispiel der U8g2 Bibliothek verwenden müssen. Wir verwenden das Beispiel *GraphicsTest*.

Dieses Beispiel finden Sie unter **Datei** → **Beispiele** → **U8g2** → **full_buffer** → **GraphicsTest**. In diesem Code müssen Sie die Kommentarstriche in Zeile 184 entfernen, bevor Sie diese mittels der Taste **Hochladen** ausführen können.

Die Zeile 184 enthält nämlich in diesem Falle die Deklaration des Displays mit 8080. Sie können die anderen Beispiele natürlich genauso verwenden, indem Sie in derselben Zeile in den anderen Codebeispielen auch die Kommentarstriche entfernen.

```

180 //U8G2_ST7920_192X32_F_8080 u8g2(U8G2_R0, 8, 9, 10, 11, 4, 5, 6, 7, /*enable=*/ 18, /*cs=*/ U8X8_PIN_NONE, /*dc=*/ 17, /*reset=*/ U8X8_PIN_NONE);
181 //U8G2_ST7920_192X32_F_SW_SPI u8g2(U8G2_R0, /*clock=*/ 18 /* A4 */, /*data=*/ 16 /* A2 */, /*CS=*/ 17 /* A3 */, /*reset=*/ U8X8_PIN_NONE);
182 //U8G2_ST7920_160X32_F_8080 u8g2(U8G2_R0, 8, 9, 10, 11, 4, 5, 6, 7, /*enable=*/ 18, /*cs=*/ U8X8_PIN_NONE, /*dc=*/ 17, /*reset=*/ U8X8_PIN_NONE);
183 //U8G2_ST7920_160X32_F_SW_SPI u8g2(U8G2_R0, /*clock=*/ 18 /* A4 */, /*data=*/ 16 /* A2 */, /*CS=*/ 17 /* A3 */, /*reset=*/ U8X8_PIN_NONE);
184 U8G2_ST7920_128X64_F_8080 u8g2(U8G2_R0, 8, 9, 10, 11, 4, 5, 6, 7, /*enable=*/ 18 /* A4 */, /*cs=*/ U8X8_PIN_NONE, /*dc/rs=*/ 17 /* A3 */, /*reset=*/ 15 /* A1 */); // Remember to set R/W to 0
185 //U8G2_ST7920_128X64_F_SW_SPI u8g2(U8G2_R0, /*clock=*/ 18 /* A4 */, /*data=*/ 16 /* A2 */, /*CS=*/ 17 /* A3 */, /*reset=*/ U8X8_PIN_NONE);
186 //U8G2_ST7920_128X64_F_SW_SPI u8g2(U8G2_R0, /*clock=*/ 13, /*data=*/ 11, /*CS=*/ 10, /*reset=*/ 8);

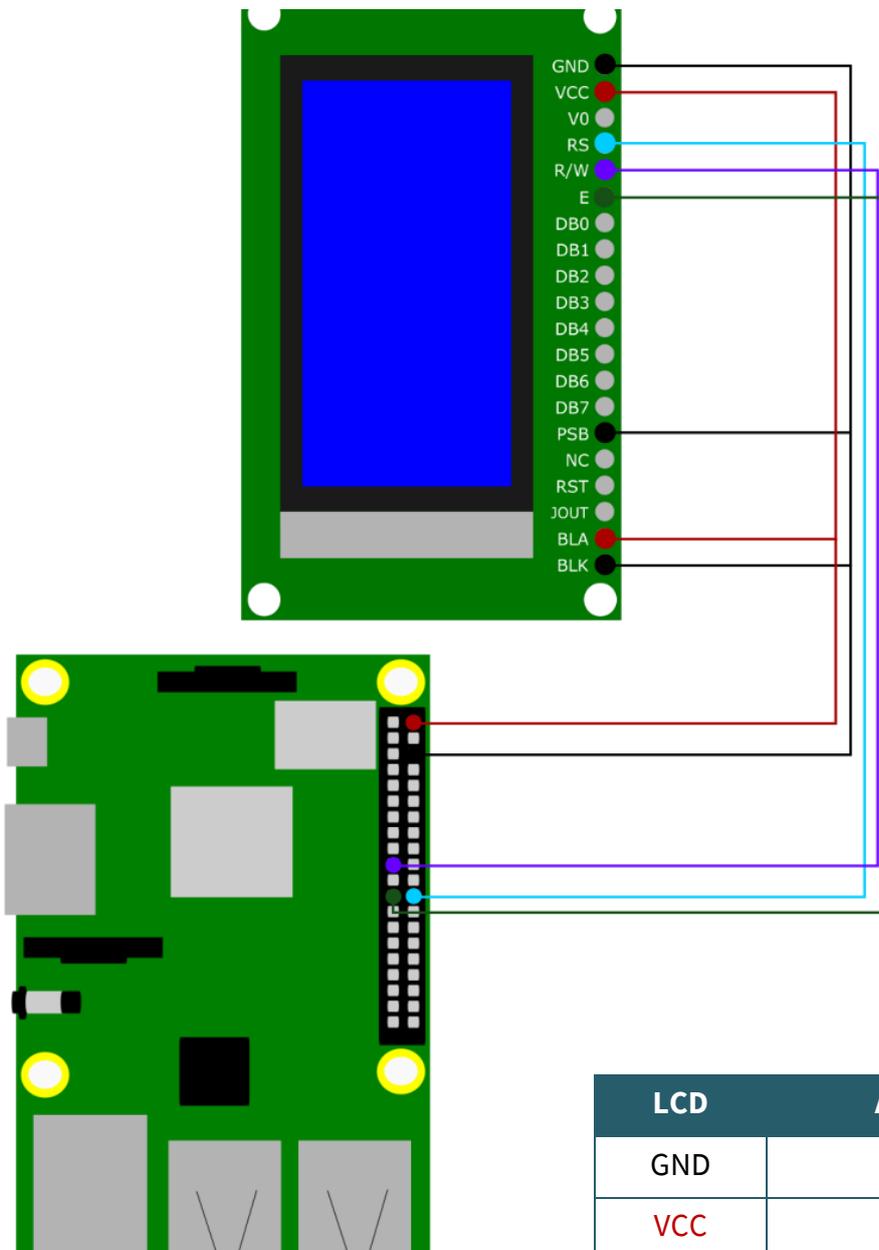
```

Beachten Sie, dass unter **Werkzeuge Board** und **Port** richtig ausgewählt sein müssen.

4. VERWENDUNG MIT DEM RASPBERRY PI

Wir verwenden das LCD mit dem Raspberry Pi nur mittels SPI.

4.1 Anschluss



LCD	Arduino
GND	GND
VCC	5V
RS	GPIO 8
R/W	GPIO 10
E	GPIO 11
PSB	GND
BLA	5V
BLK	GND

4.2 Bibliothek

Wir verwenden die Bibliothek [u8g2-rpi-demo](#) von [ribasco](#), welche unter der [Apache License 2.0](#) veröffentlicht wurde.

Laden Sie sich zunächst dafür **git** herunter. Dazu geben Sie die folgenden Befehle ein:

```
sudo apt-get update
```

```
sudo apt-get install git
```

Führen Sie im Anschluss den folgenden Befehl aus, um die Bibliothek herunterzuladen.

```
git clone https://github.com/ribasco/u8g2-rpi-demo.git
```

Geben Sie nun die folgenden Befehle in der Konsole ein, damit alle nötigen Dateien gedownloadet und installiert werden.

Manche Schritte in der Installation könnten Zeit in Anspruch nehmen.

```
cd u8g2-rpi-demo/
```

```
sudo apt-get install libcurl4-openssl-dev
```

```
cd ..
```

```
git clone https://github.com/Kitware/CMake.git
```

```
cd CMake
```

```
./bootstrap --system-curl
```

```
wget https://project-downloads.drogon.net/wiringpi-latest.deb
```

```
sudo dpkg -i wiringpi-latest.deb
```

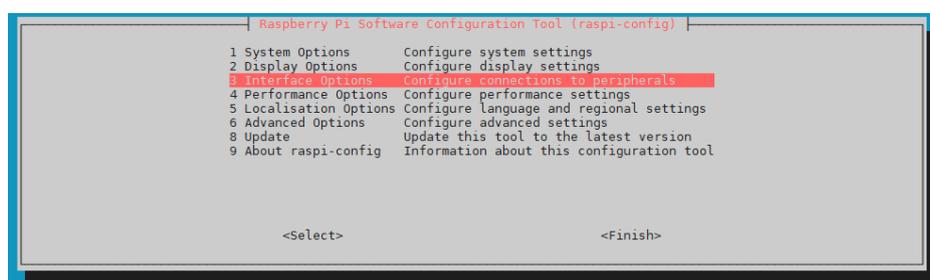
```
make
```

```
sudo make install
```

Aktivieren Sie nun die serielle Kommunikation, indem Sie die Konfiguration des Raspberry Pis mit dem folgenden Befehl aufrufen.

```
sudo raspi-config
```

Wählen Sie dort unter **3 Interface Options** → **P4 SPI** aus.



```
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 using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell 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>
```

Stimmen Sie der Frage „*Would you like the SPI interface to be enabled?*“ mit „**Yes**“ zu.

```
Would you like the SPI interface to be enabled?

<Yes>                                <No>
```

Schießen Sie die Aktivierung der SPI Schnittstelle mit „**OK**“ ab und beenden Sie die Konfiguration mit „**Finish**“.

```
The SPI interface is enabled

<Ok>
```

Sie haben nun erfolgreich die SPI-Schnittstelle aktiviert. Bitte starten Sie Ihren Raspberry Pi jetzt neu.

```
sudo reboot
```

4.3 Verwendung der Bibliothek

Sie können das Beispielprogramm der Bibliothek mit dem folgenden Befehl kompilieren.

```
cd ~/u8g2-rpi-demo/
```

```
./compile.sh
```

Sie können mit **Y** bestätigen, dass der Beispielcode gestartet wird oder Sie führen den folgenden Befehl aus.

```
./build/arm/debug/bin/u8g2demo
```

Sie können auch den Beispielcode mittels der *main.cpp* editieren. Wenn Sie diese Datei bearbeiten, müssen Sie diese erneut mit dem ersten Befehl kompilieren.

5. 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.

6. 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 98469 – 66 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net