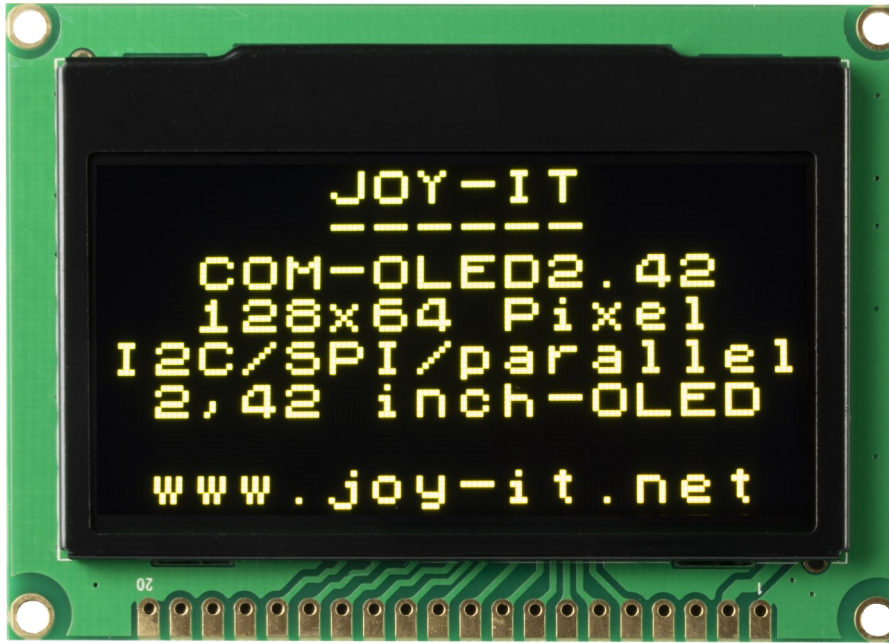


OLED-DISPLAY MODULE

COM-OLED2.42



1. GENERAL INFORMATION

Dear customer,
thank you very much for choosing our product.
In following, we will introduce you to what to observe while starting up
and using this product.
Should you encounter any unexpected problems during use, please do
not hesitate to contact us.

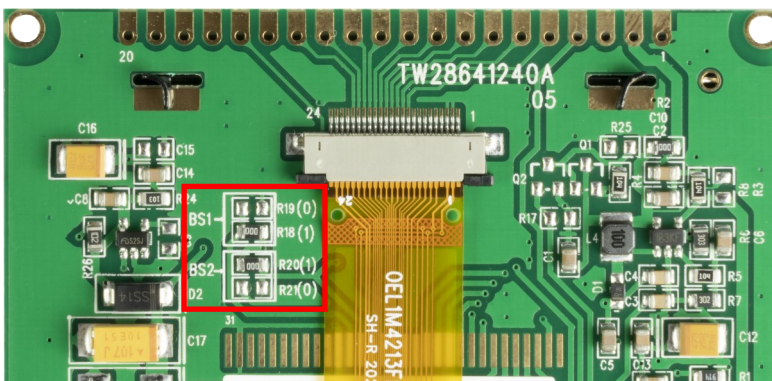
2. SETUP OF THE DISPLAY INTERFACE

The display can be controlled in 4 different ways, via I2C, SPI, 8-bit parallel 6800 interface and 8-bit parallel 8080 interface.

The display is delivered pre-configured for control via SPI. If you want to use one of the other control methods, you have to re-solder the resistors BS1 and BS2 on the back of the board.

In the table, you can see how the resistors must be set for the respective mode.

	6800-parallel	8080-parallel	I2C	SPI
BS1	0	1	1	0
BS2	1	1	0	0



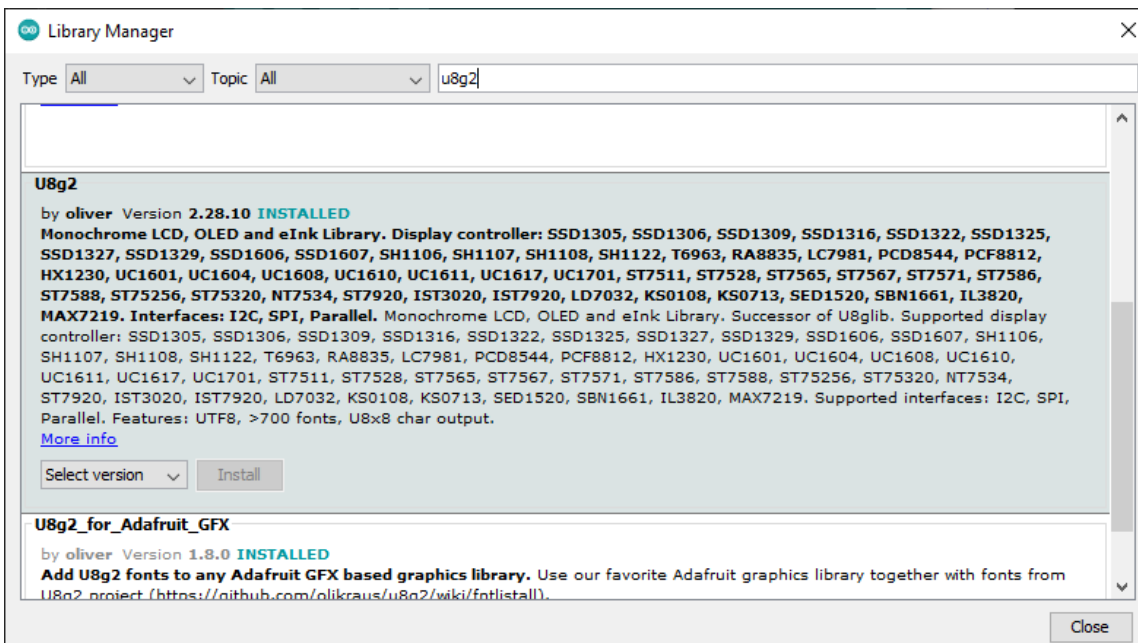
3. USE WITH AN ARDUINO

As the display works with a 3V logic level and most Arduinos with 5V, we use an Arduino Pro Mini 3.3V in this example.

If you want to use an Arduino with a 5V logic level, such as an Arduino Uno, you have to reduce all data lines leading from the Arduino to the display from 5V to 3.3V with a logic level converter.

First you need to install the required library in your Arduino IDE.

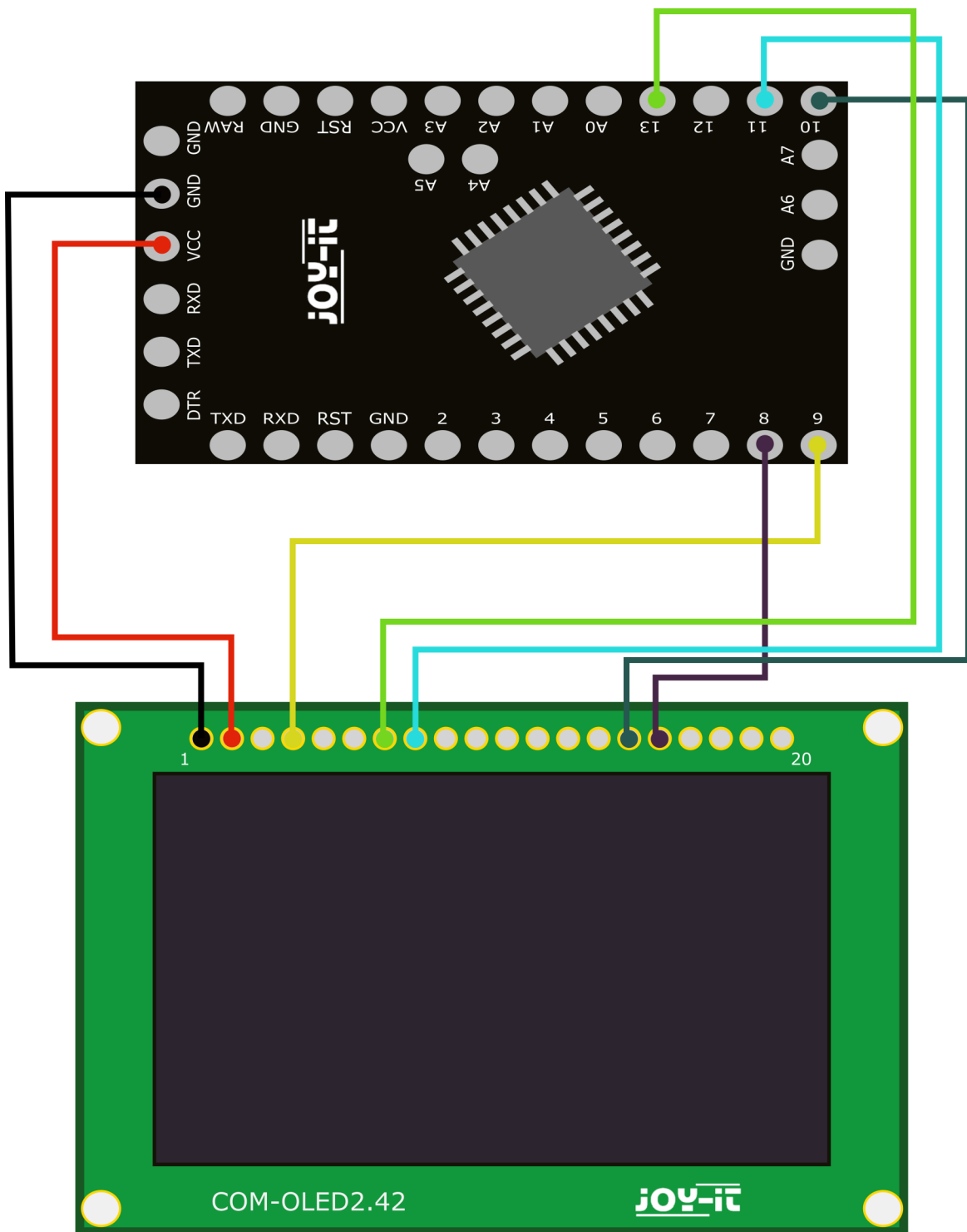
To do this, go to **Tools -> Manage Libraries...** Search for **u8g2** and install the library **U8g2 by oliver**



SPI-Interface

Wiring

Display Pin	1	2	4	7	8	15	16
Arduino Pro Mini Pin	GND	3,3V (VCC)	9	13	11	10	8



SPI-Interface

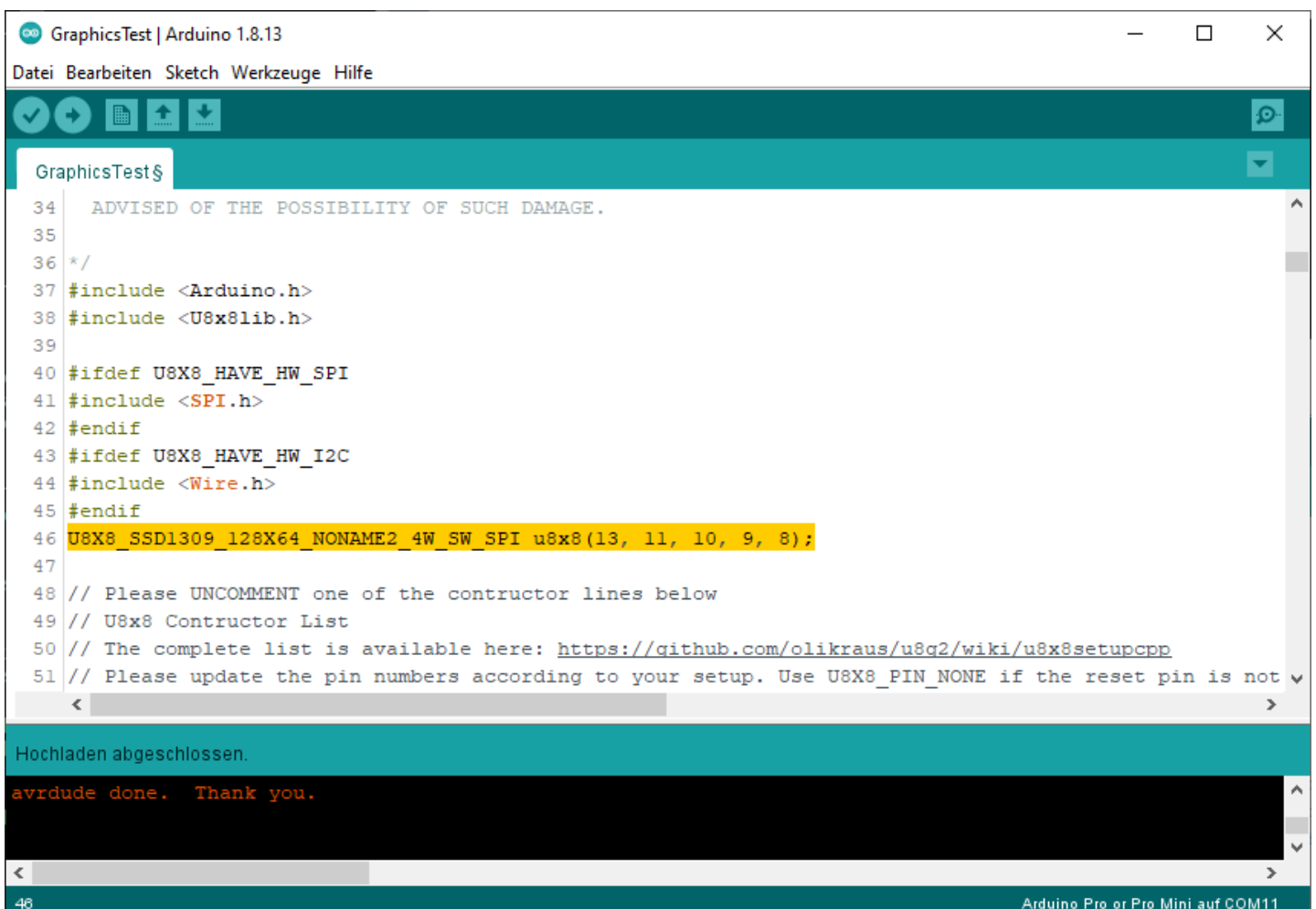
Now open the GraphicTest code sample of the library.

To do this, click on:

File -> Examples -> U8g2 -> u8x8-> GraphicTest

Now insert the following constructor for the display into the programme, as shown in the picture below:

```
U8X8_SSD1309_128X64_NONAME2_4W_SW_SPI u8x8(13, 11, 10, 9, 8);
```



The screenshot shows the Arduino IDE interface for the 'GraphicsTest' sketch. The code editor displays the following code:

```
34 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
35
36 */
37 #include <Arduino.h>
38 #include <U8x8lib.h>
39
40 #ifdef U8X8_HAVE_HW_SPI
41 #include <SPI.h>
42 #endif
43 #ifdef U8X8_HAVE_HW_I2C
44 #include <Wire.h>
45 #endif
46 U8X8_SSD1309_128X64_NONAME2_4W_SW_SPI u8x8(13, 11, 10, 9, 8);
47
48 // Please UNCOMMENT one of the constructor lines below
49 // U8x8 Constructor List
50 // The complete list is available here: https://github.com/olikraus/u8g2/wiki/u8x8setupcpp
51 // Please update the pin numbers according to your setup. Use U8X8_PIN_NONE if the reset pin is not
```

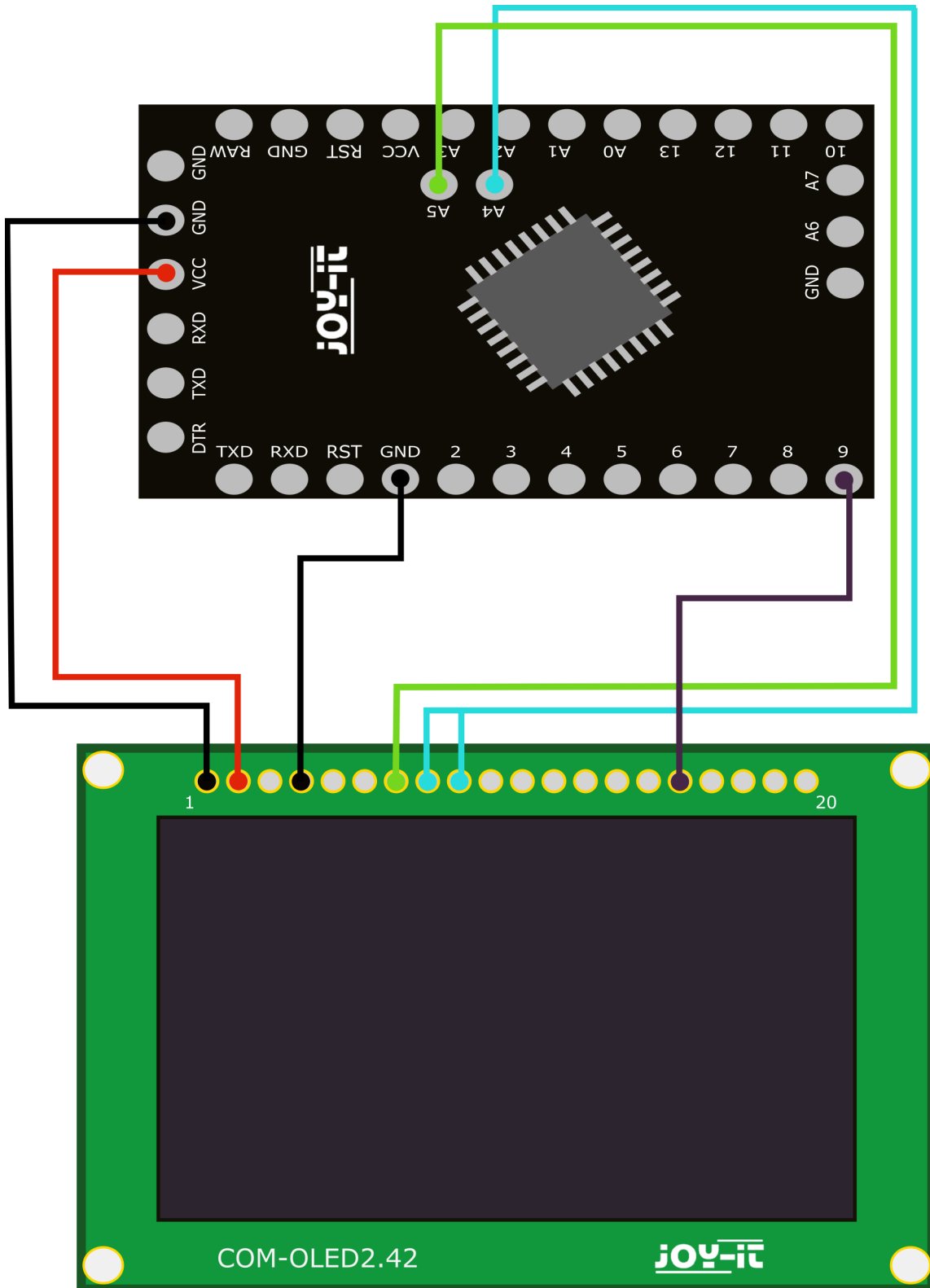
The line `U8X8_SSD1309_128X64_NONAME2_4W_SW_SPI u8x8(13, 11, 10, 9, 8);` is highlighted in yellow. Below the code editor, the status bar shows 'Hochladen abgeschlossen.' and 'avrdude done. Thank you.' The bottom status bar indicates '48' and 'Arduino Pro or Pro Mini auf COM11'.

Now you can upload the example to your Arduino.

I2C-Interface

Wiring

Display Pin	1	2	4	7	8	9	16
Arduino Pro Mini Pin	GND	3,3V (VCC)	GND	A5	A4	A4	9



I2C-Interface

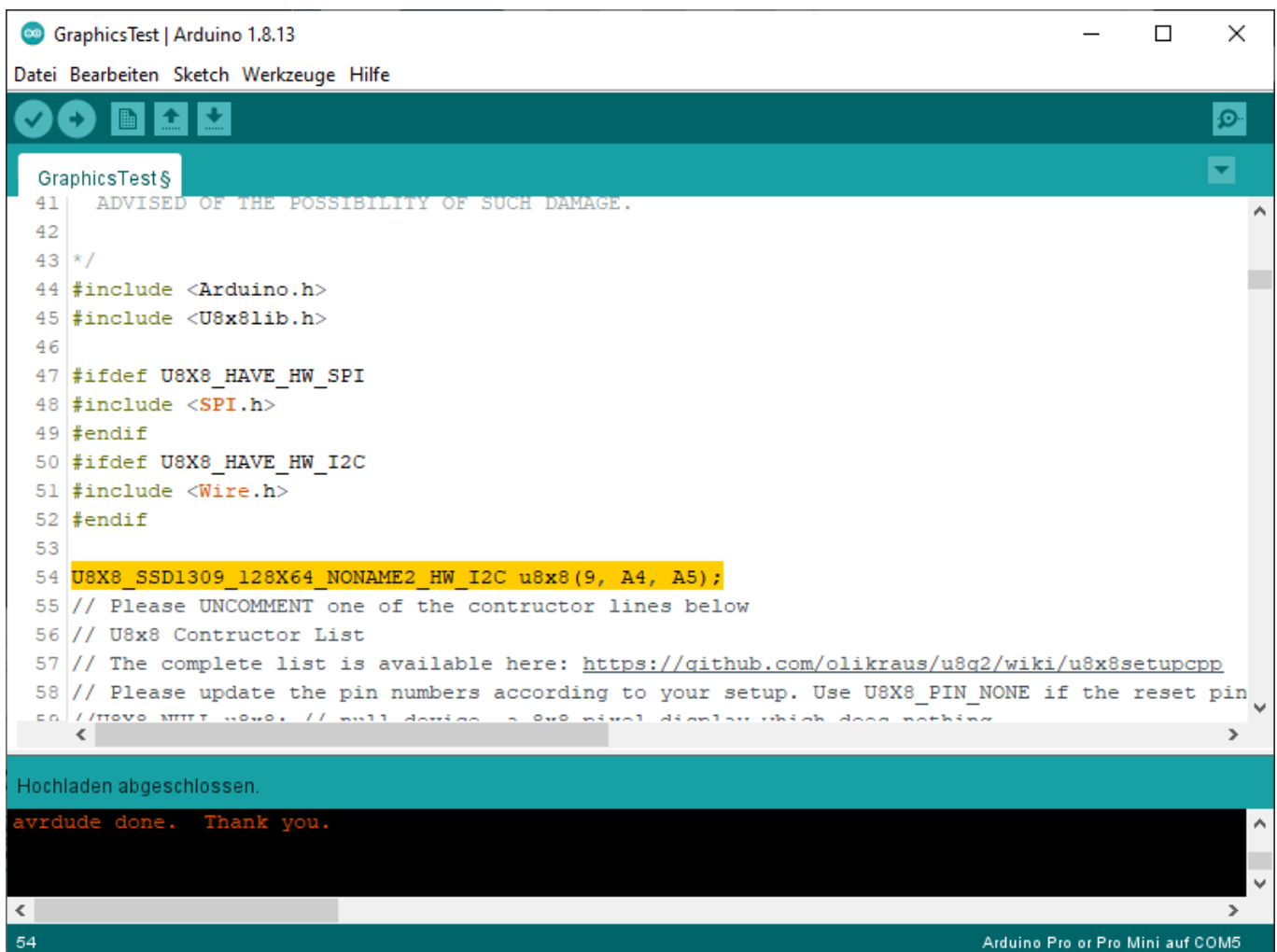
Now open the GraphicTest code sample of the library.

To do this, click on:

File -> Examples -> U8g2 -> u8x8-> GraphicTest

Now insert the following constructor for the display into the programme, as shown in the picture below:

```
U8X8_SSD1309_128X64_NONAME2_HW_I2C u8x8(9, A4, A5);
```



The screenshot shows the Arduino IDE interface for the 'GraphicsTest' sketch. The code editor displays the following code:

```
GraphicsTest$
41  ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
42
43  */
44  #include <Arduino.h>
45  #include <U8x8lib.h>
46
47  #ifdef U8X8_HAVE_HW_SPI
48  #include <SPI.h>
49  #endif
50  #ifdef U8X8_HAVE_HW_I2C
51  #include <Wire.h>
52  #endif
53
54  U8X8_SSD1309_128X64_NONAME2_HW_I2C u8x8(9, A4, A5);
55  // Please UNCOMMENT one of the constructor lines below
56  // U8x8 Constructor List
57  // The complete list is available here: https://github.com/olikraus/u8g2/wiki/u8x8setupcpp
58  // Please update the pin numbers according to your setup. Use U8X8_PIN_NONE if the reset pin
59  // does not exist. // null device = 8x8 pixel display which does nothing
```

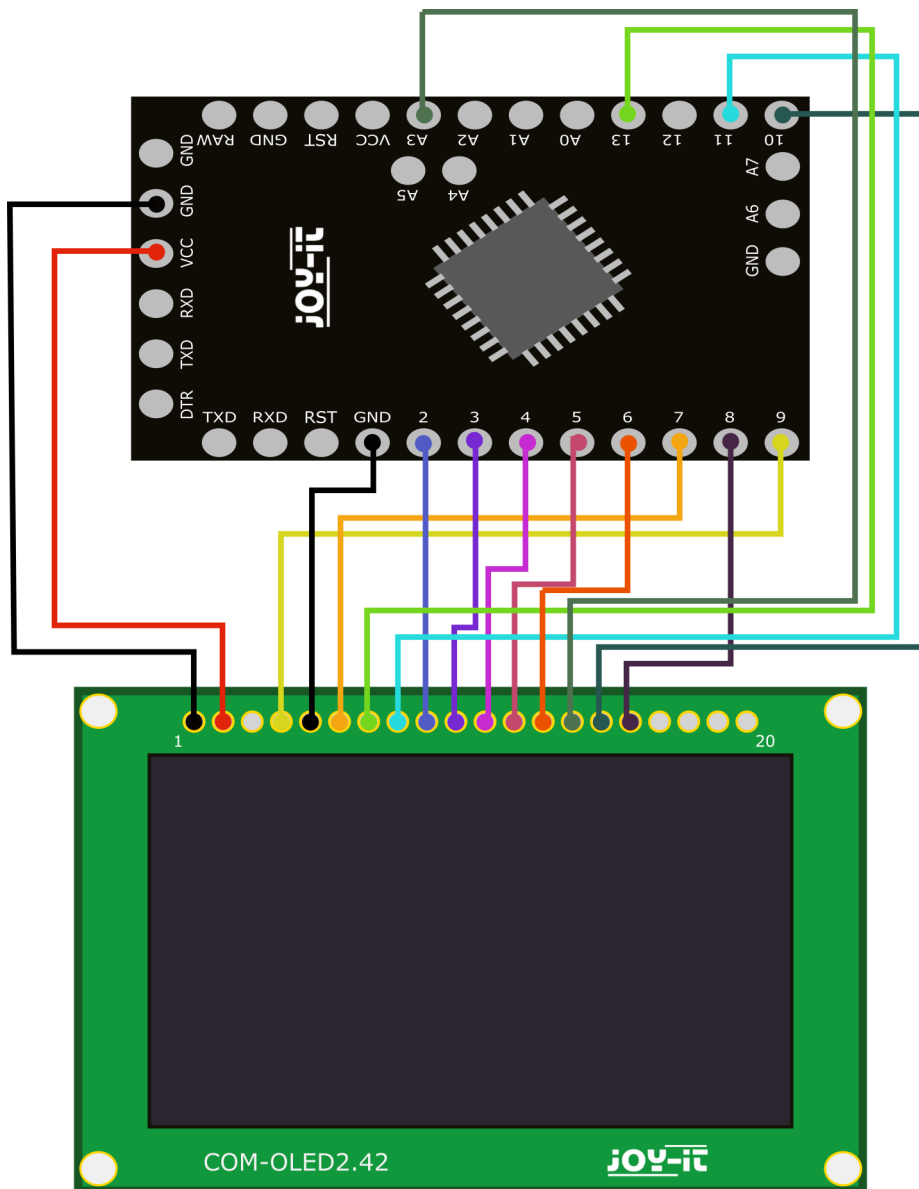
The line `U8X8_SSD1309_128X64_NONAME2_HW_I2C u8x8(9, A4, A5);` is highlighted in yellow. Below the code editor, a status bar shows 'Hochladen abgeschlossen.' and 'avrdude done. Thank you.' The bottom status bar indicates '54' and 'Arduino Pro or Pro Mini auf COM5'.

Now you can upload the example to your Arduino.

8 bit Parallel 6800-Interface

Wiring

Display Pin	1	2	4	5	6	7	8	9	10	11	12	13	14	15	16
Arduino Pro Mini Pin	GND	3,3V (VCC)	9	GND	7	13	11	2	3	4	5	6	A3	10	8



8 bit Parallel 6800-Interface

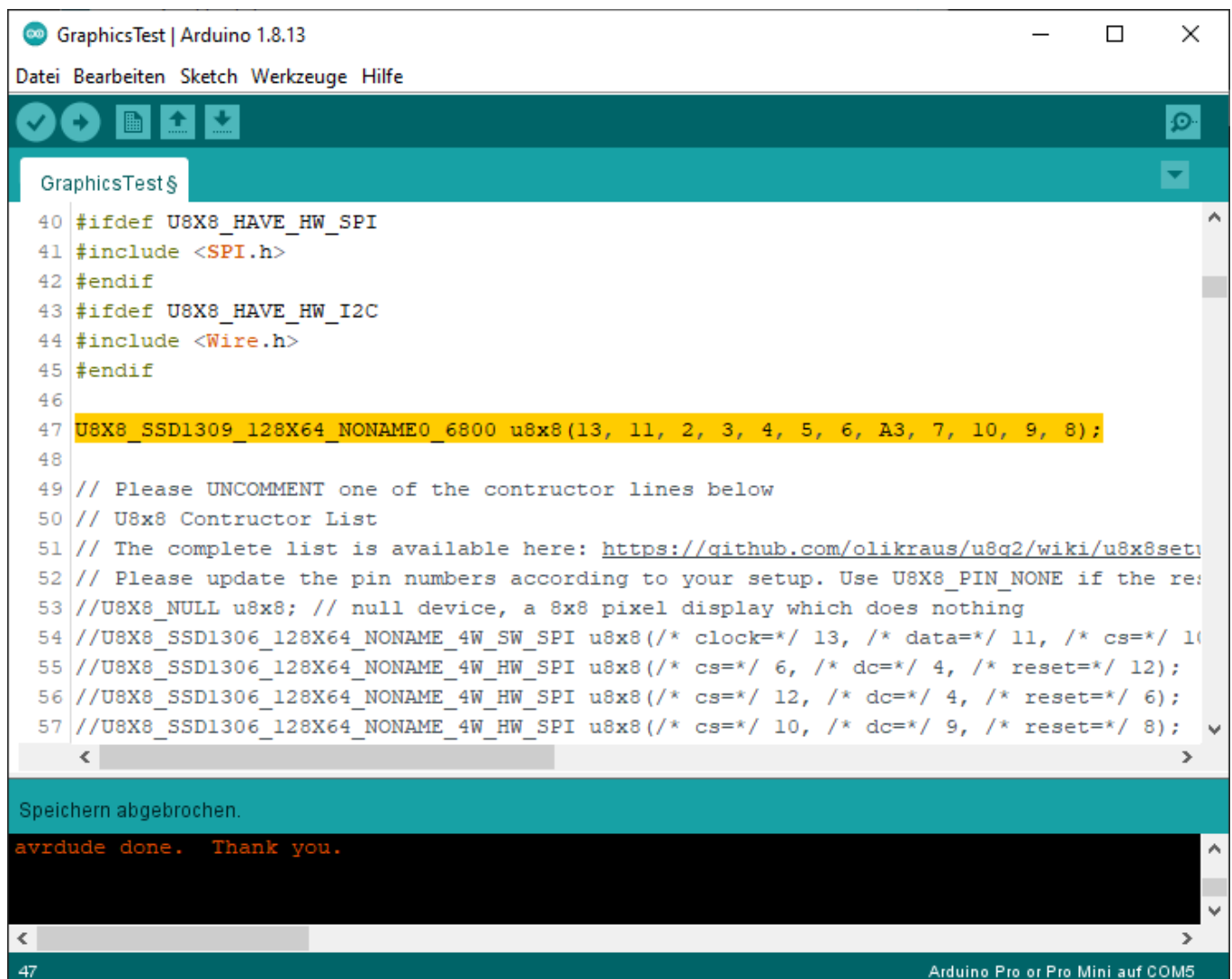
Now open the GraphicTest code sample of the library.

To do this, click on:

File -> Examples -> U8g2 -> u8x8-> GraphicTest

Now insert the following constructor for the display into the programme, as shown in the picture below:

```
U8X8_SSD1309_128X64_NONAME0_6800 u8x8(13, 11, 2, 3, 4, 5, 6,
A3, 7, 10, 9, 8);
```



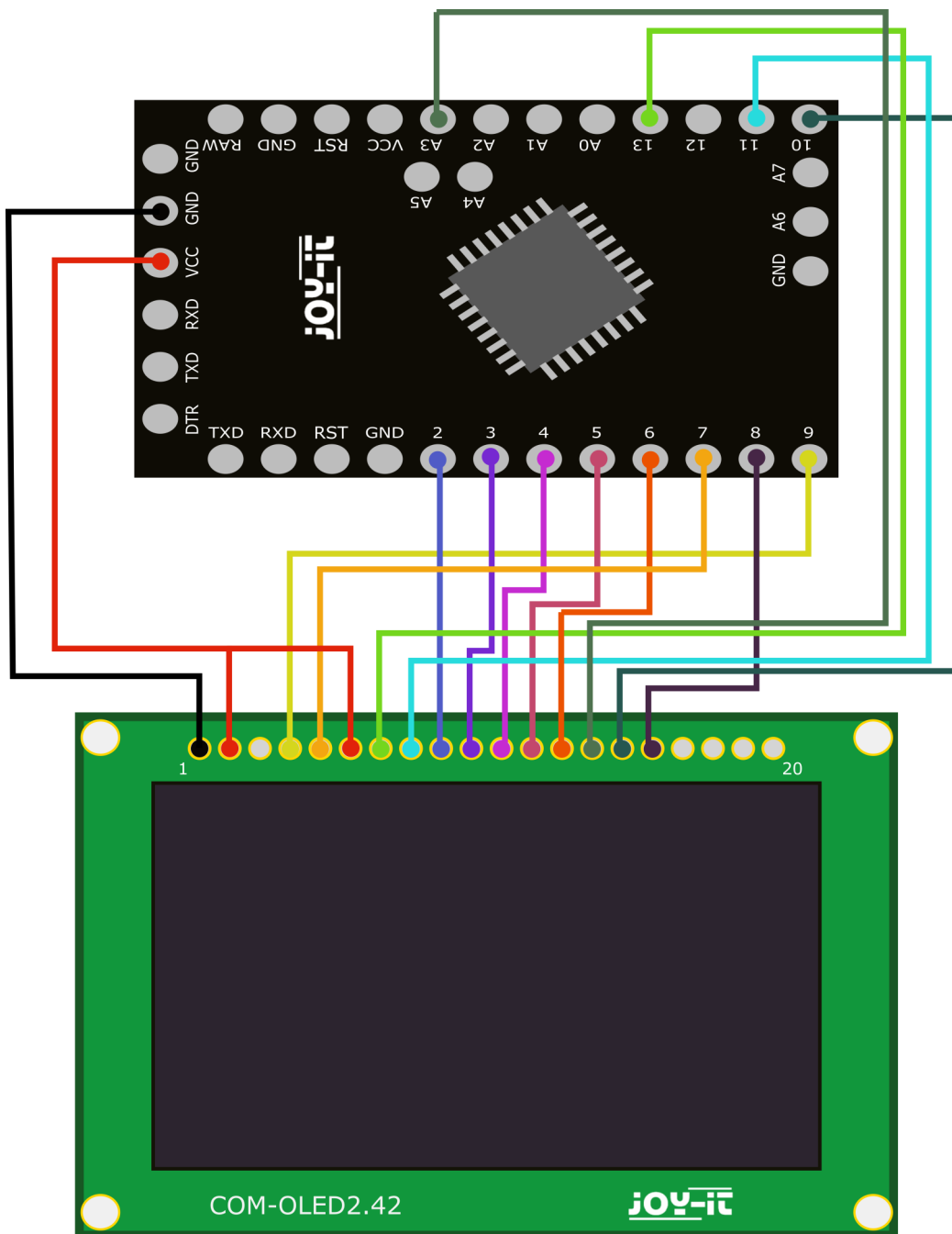
```
GraphicsTest | Arduino 1.8.13
Datei Bearbeiten Sketch Werkzeuge Hilfe
GraphicsTest$
40 #ifndef U8X8_HAVE_HW_SPI
41 #include <SPI.h>
42 #endif
43 #ifndef U8X8_HAVE_HW_I2C
44 #include <Wire.h>
45 #endif
46
47 U8X8_SSD1309_128X64_NONAME0_6800 u8x8(13, 11, 2, 3, 4, 5, 6, A3, 7, 10, 9, 8);
48
49 // Please UNCOMMENT one of the constructor lines below
50 // U8x8 Constructor List
51 // The complete list is available here: https://github.com/olikraus/u8g2/wiki/u8x8set
52 // Please update the pin numbers according to your setup. Use U8X8_PIN_NONE if the res
53 //U8X8_NULL u8x8; // null device, a 8x8 pixel display which does nothing
54 //U8X8_SSD1306_128X64_NONAME_4W_SW_SPI u8x8(/* clock=*/ 13, /* data=*/ 11, /* cs=*/ 10
55 //U8X8_SSD1306_128X64_NONAME_4W_HW_SPI u8x8(/* cs=*/ 6, /* dc=*/ 4, /* reset=*/ 12);
56 //U8X8_SSD1306_128X64_NONAME_4W_HW_SPI u8x8(/* cs=*/ 12, /* dc=*/ 4, /* reset=*/ 6);
57 //U8X8_SSD1306_128X64_NONAME_4W_HW_SPI u8x8(/* cs=*/ 10, /* dc=*/ 9, /* reset=*/ 8);
Speichern abgebrochen.
avrdude done. Thank you.
47 Arduino Pro or Pro Mini auf COM5
```

Now you can upload the example to your Arduino.

8 bit Parallel 8080-Interface

Wiring

Display Pin	1	2	4	5	6	7	8	9	10	11	12	13	14	15	16
Arduino Pro	GND	3,3V	9	7	3,3V	13	11	2	3	4	5	6	A3	10	8



8 bit Parallel 8080-Interface

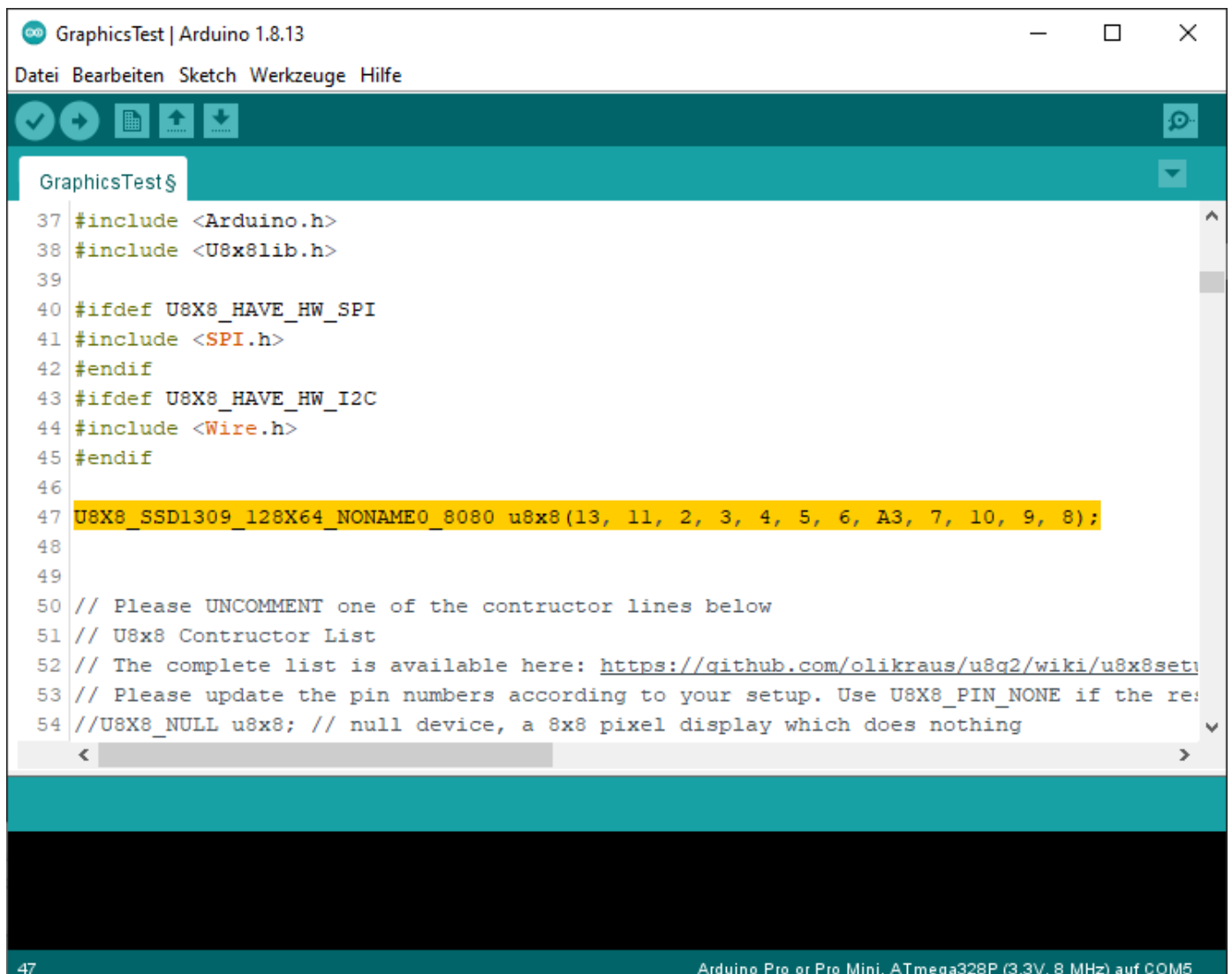
Now open the GraphicTest code sample of the library.

To do this, click on:

File -> Examples -> U8g2 -> u8x8-> GraphicTest

Now insert the following constructor for the display into the programme,

```
U8X8_SSD1309_128X64_NONAME0_8080 u8x8(13, 11, 2, 3, 4, 5, 6,
A3, 7, 10, 9, 8);
```



```
GraphicsTest | Arduino 1.8.13
Datei Bearbeiten Sketch Werkzeuge Hilfe
GraphicsTest$
37 #include <Arduino.h>
38 #include <U8x8lib.h>
39
40 #ifdef U8X8_HAVE_HW_SPI
41 #include <SPI.h>
42 #endif
43 #ifdef U8X8_HAVE_HW_I2C
44 #include <Wire.h>
45 #endif
46
47 U8X8_SSD1309_128X64_NONAME0_8080 u8x8(13, 11, 2, 3, 4, 5, 6, A3, 7, 10, 9, 8);
48
49
50 // Please UNCOMMENT one of the constructor lines below
51 // U8x8 Constructor List
52 // The complete list is available here: https://github.com/olikraus/u8g2/wiki/u8x8set
53 // Please update the pin numbers according to your setup. Use U8X8_PIN_NONE if the res
54 //U8X8_NULL u8x8; // null device, a 8x8 pixel display which does nothing
47 Arduino Pro or Pro Mini, ATmega328P (3.3V, 8 MHz) auf COM5
```

Now you can upload the example to your Arduino.

4. USE WITH THE RASPBERRY PI

To make using the display with the Raspberry Pi particularly easy, we use the `luma.oled` library.

You can install the dependencies required for installation with the following commands:

```
sudo apt-get update
```

```
sudo apt-get install git python3 python3-dev python3-pip python3-pip python3-pil libjpeg-dev zlib1g-dev libfreetype6-dev liblcms2-dev libopenjp2-7 libtiff5 build-essential libsdl-dev libportmidi-dev libsdl-ttf2.0-dev libsdl-mixer1.2-dev libsdl-image1.2-dev
```

First, we install the library and download the sample files, this is done with the following commands:

```
sudo -H pip3 install --upgrade luma.oled
```

```
sudo git clone https://github.com/rm-hull/luma.examples.git
```

Next, you must give the user (in this case "pi") permission to access the required hardware. If you are not using the user "pi", change the user at the end of the command accordingly.

```
sudo usermod -a -G spi,gpio,i2c pi
```

Then restart your Raspberry Pi with the following command:

```
sudo reboot
```

After the reboot, enter following command into the terminal:

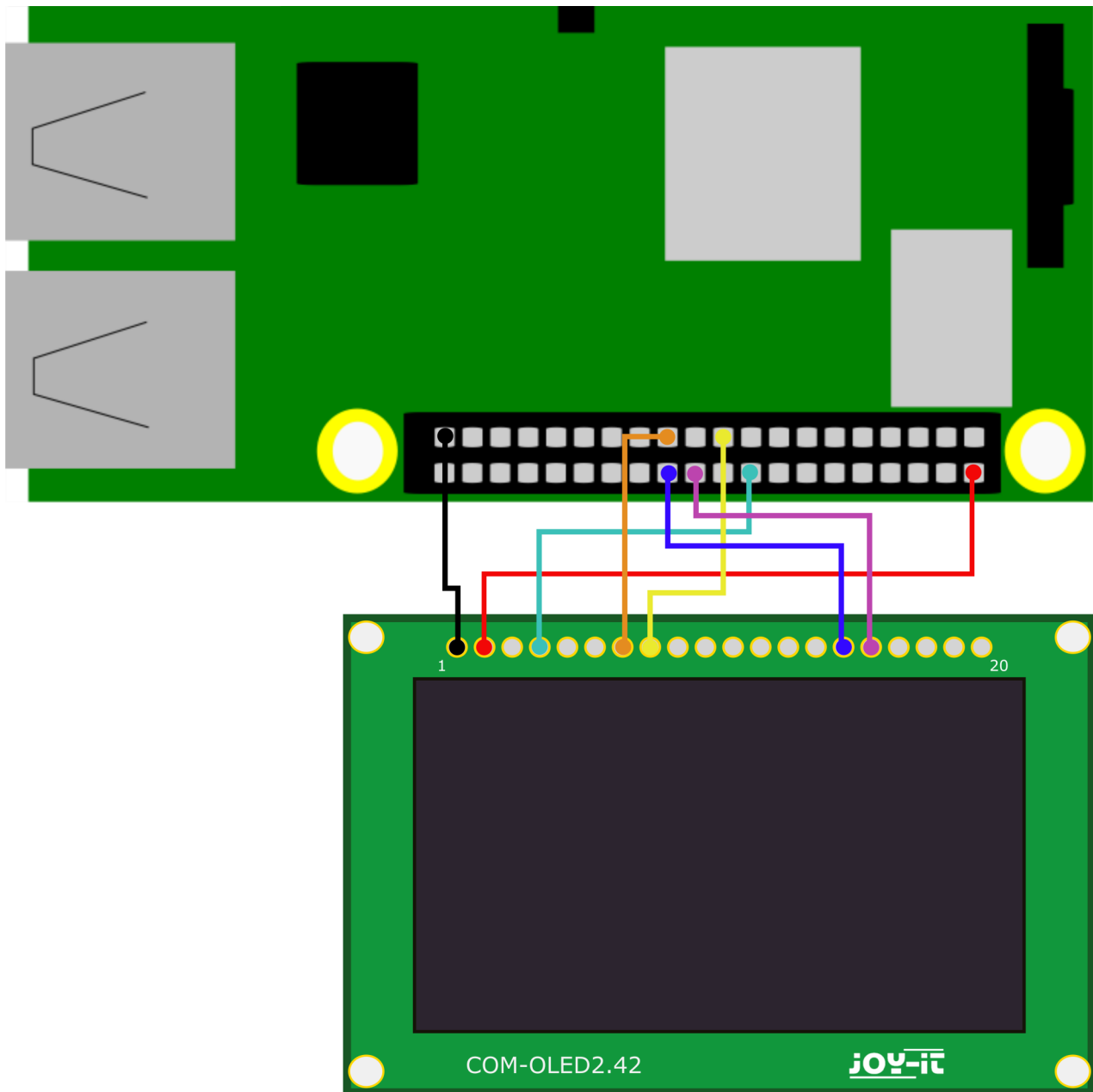
```
sudo raspi-config
```

There you can now activate **SPI** and **I2C** under **3 Interface Options** so that you can use both interfaces.

SPI-Interface

Wiring

Display Pin	1	2	4	7	8	15	16
Raspberry Pin	GND	5V	Pin 18	Pin 23	Pin 19	Pin 24	Pin 22



After you have connected the display, you can execute a sample programme with the following two commands:

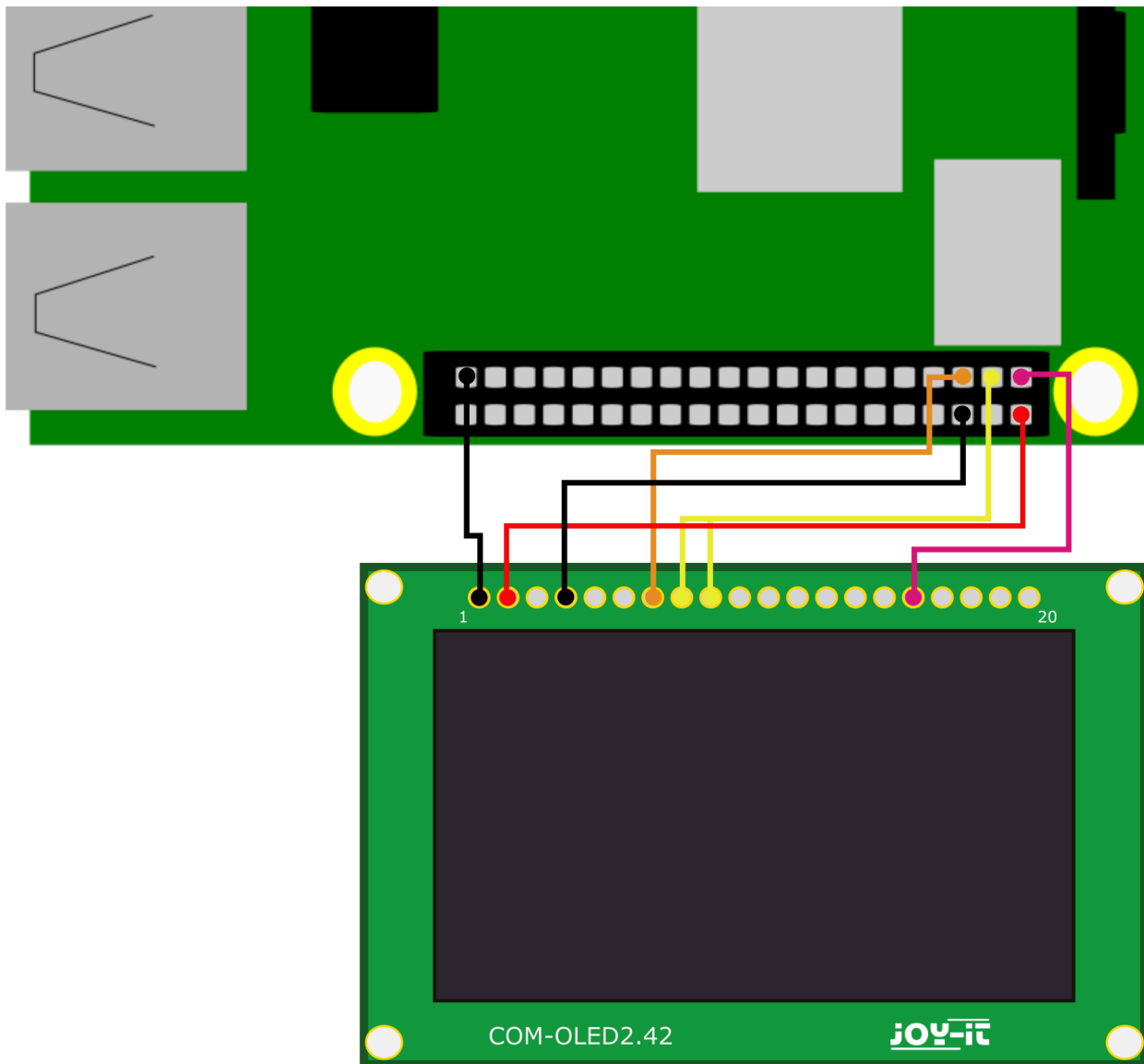
```
cd ~/luma.examples/examples/
```

```
sudo python3 demo.py -i spi
```

I2C-Interface

Wiring

Display Pin	1	2	4	7	8	9	16
Raspberry Pin	GND	5V	GND	Pin 5	Pin 3	Pin 3	3,3V



After you have connected the display, you can execute a sample programme with the following two commands:

```
cd ~/luma.examples/examples/
```

```
sudo python3 demo.py
```

5. 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 fulfils 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.

6. 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 98469-66 (10-17 o'clock)

For further information please visit our website:

www.joy-it.net