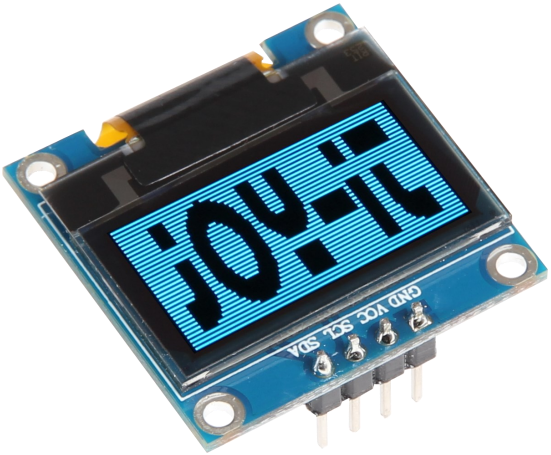
SBC-OLED01

OLED-Display 128x64 i2c



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.

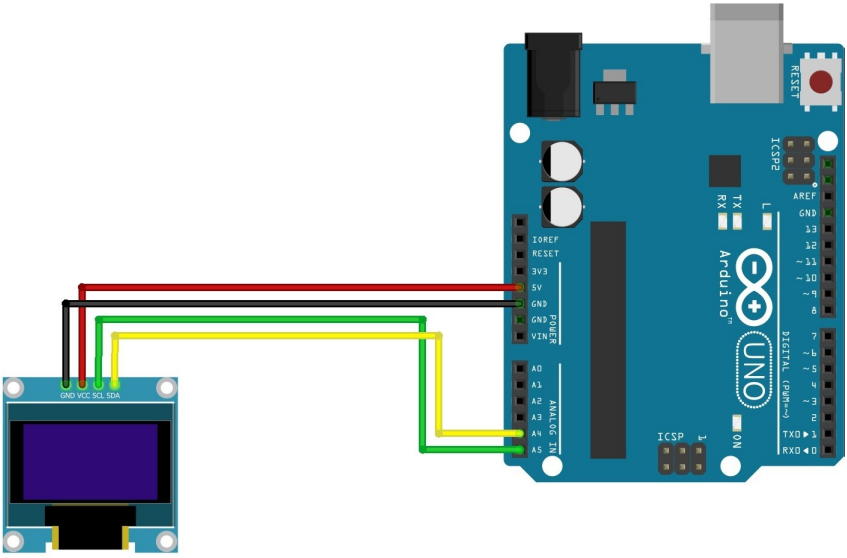
Should you encounter any unexpected problems, do not hesitate to contact us.

INDEX

1. USAGE WITH THE ARDUINO

Connecting the display

Connect the display the display like shown in the picture with the pins of your Arduino Uno.



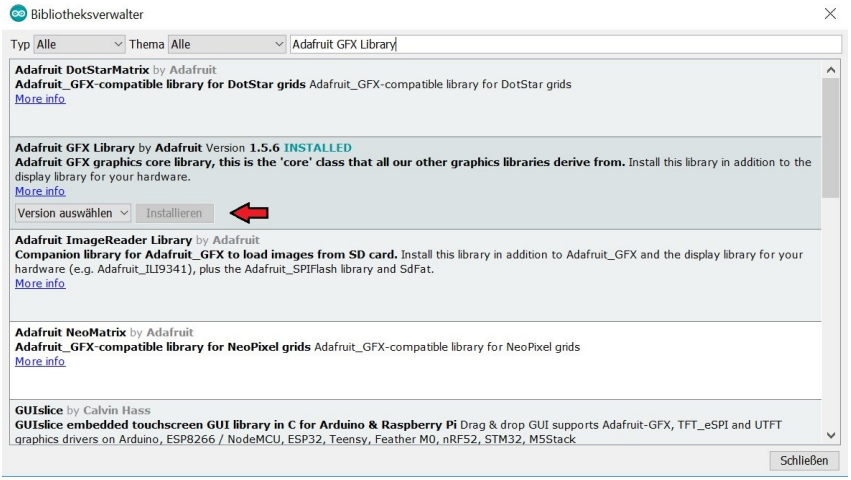
SBC-OLED01	Arduino Uno
GND	GND
VCC	5V or 3.3V
SCL	A5
SDA	A4

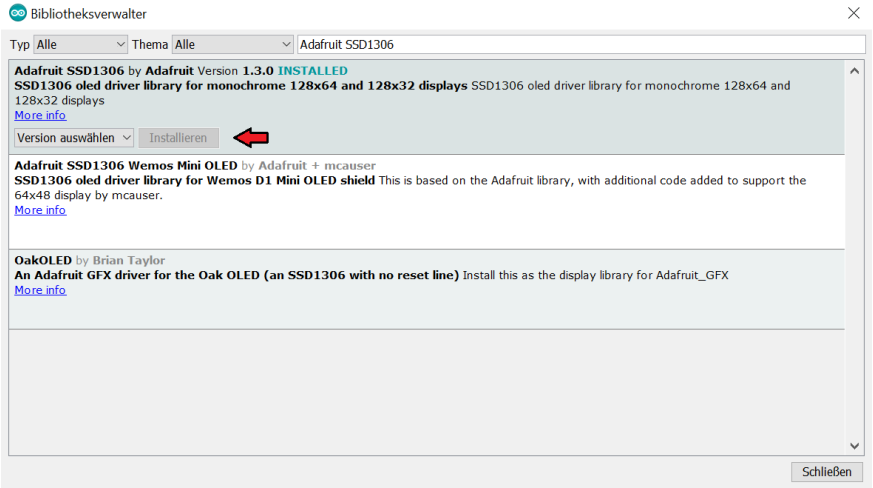
Installation of the libraries

Before you can transfer the code example to your Arduino, you must install the additional libraries Adafruit GFX Library and Adafruit SSD1306.
These libraries allow you to use the display as fast and easy as possible.

First, open the library management in your Arduino software.

Search in the opening library management for the libraries **Adafruit GFX Library** and **Adafruit SSD1306** and install them.





As soon as the libraries are installed, the installation of the display is completed.
In the next step, you can transfer our code examples to test the display.

Code examples

In the following, you can use the code example to use the display.

Copy this code completely to your Arduino.

Ensure that the correct board and port are selected in the section **tools** in the menu.

Code example 1 (text)

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

void setup()   {

     display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //
Adressierung beachten, hier 0x3C!
}

int i;

void loop() {
     display.clearDisplay();
     display.setTextColor(WHITE);
     display.setTextSize(1);
     display.setCursor(23,0);
     display.println("OLED - Display");
     display.setCursor(23,12);
     display.println("www.joy-it.net");
     display.setCursor(36,24);
     display.println("SBC-OLED01");
     display.display();
     delay(8000);
     display.clearDisplay();
     display.invertDisplay(true);
     delay(8000);
     display.invertDisplay(false);
     delay(1000);
}
```

Code example 2 (graphic)

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

#if (SSD1306_LCDHEIGHT != 64)
#endif

const unsigned char logo [] PROGMEM = {
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe3, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0xfe, 0x00, 0x3f, 0xf8, 0x0f,
0xc0, 0x7f, 0xff, 0xf8, 0x0f, 0x00, 0x00, 0x7f, 0xff, 0xf8, 0x0f, 0xfc, 0x00, 0x0f,
0xf8, 0x0f, 0xc0, 0x7f, 0xff, 0xf8, 0x0f, 0x00, 0x00, 0x7f, 0xff, 0xf8, 0x0f, 0xf0,
0x00, 0x03, 0xf8, 0x0f, 0xc0, 0x7f, 0xff, 0xf8, 0x0f, 0x00, 0x00, 0x7f, 0xff, 0xf8,
0x0f, 0xe0, 0x00, 0x01, 0xf8, 0x0f, 0xc0, 0x7f, 0xff, 0xf8, 0x0f, 0x00, 0x00, 0x7f,
0xff, 0xf8, 0x0f, 0xc0, 0x00, 0x00, 0xf8, 0x0f, 0xc0, 0x7f, 0xff, 0xf8, 0x0f, 0x00,
0x00, 0x7f, 0xff, 0xf8, 0x0f, 0xc0, 0x1c, 0x00, 0xf8, 0x0f, 0xc0, 0x7f, 0xff, 0xf8,
0x0f, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x80, 0x7f, 0x00, 0x78, 0x0f, 0xc0, 0x7f,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0xff, 0x80, 0x78, 0x0f,
0xc0, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0xff, 0xc0,
0x38, 0x0f, 0xc0, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x01,
0xff, 0xc0, 0x38, 0x07, 0x80, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8,
0x0f, 0x01, 0xff, 0xe0, 0x3c, 0x00, 0x00, 0x7f, 0xff, 0xf8, 0x0f, 0x80, 0xff, 0xff,
0xff, 0xf8, 0x0f, 0x01, 0xff, 0xe0, 0x3c, 0x00, 0x00, 0xe0, 0x00, 0x18, 0x0f, 0x80,
0xff, 0xff, 0xff, 0xf8, 0x0f, 0x01, 0xff, 0xe0, 0x3e, 0x00, 0x00, 0xe0, 0x00, 0x18,
0x0f, 0x80, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0x01, 0xff, 0xe0, 0x3f, 0x00, 0x01, 0xe0,
0x00, 0x18, 0x0f, 0x80, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0x01, 0xff, 0xe0, 0x3f, 0x80,
0x03, 0xe0, 0x00, 0x18, 0x0f, 0x80, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0x01, 0xff, 0xe0,
0x3f, 0xc0, 0x0f, 0xe0, 0x00, 0x18, 0x0f, 0x80, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0x00,
0xff, 0xc0, 0x3f, 0xff, 0xff, 0xe0, 0x00, 0x18, 0x0f, 0x80, 0xff, 0xff, 0xff, 0xf8,
0x0f, 0x00, 0xff, 0xc0, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0x80, 0xff, 0xff,
0xff, 0xf0, 0x0f, 0x80, 0x7f, 0x80, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x0f, 0x80,
0x7f, 0xff, 0xfe, 0x00, 0x0f, 0x80, 0x3f, 0x00, 0x7f, 0x80, 0x07, 0xff, 0xff, 0xf8,
0x0f, 0x80, 0x00, 0xff, 0xfe, 0x00, 0x0f, 0xc0, 0x00, 0x00, 0xff, 0x80, 0x07, 0xff,
0xff, 0xf8, 0x0f, 0xc0, 0x00, 0xff, 0xfe, 0x00, 0x1f, 0xe0, 0x00, 0x01, 0xff, 0x80,
0x07, 0xff, 0xff, 0xf8, 0x0f, 0xc0, 0x00, 0xff, 0xfe, 0x00, 0x1f, 0xe0, 0x00, 0x03,
0xff, 0x80, 0x07, 0xff, 0xff, 0xf8, 0x0f, 0xc0, 0x00, 0xff, 0xfe, 0x00, 0x3f, 0xf8,
0x00, 0x07, 0xff, 0x80, 0x07, 0xff, 0xff, 0xf8, 0x0f, 0xe0, 0x00, 0xff, 0xfe, 0x00,
0x7f, 0xfc, 0x00, 0x0f, 0xff, 0x80, 0x07, 0xff, 0xff, 0xf8, 0x0f, 0xf0, 0x00, 0xff,
0xfe, 0x03, 0xff, 0xff, 0x80, 0x7f, 0xff, 0x80, 0x07, 0xff, 0xff, 0xf8, 0x0f, 0xfe,
0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff };
```

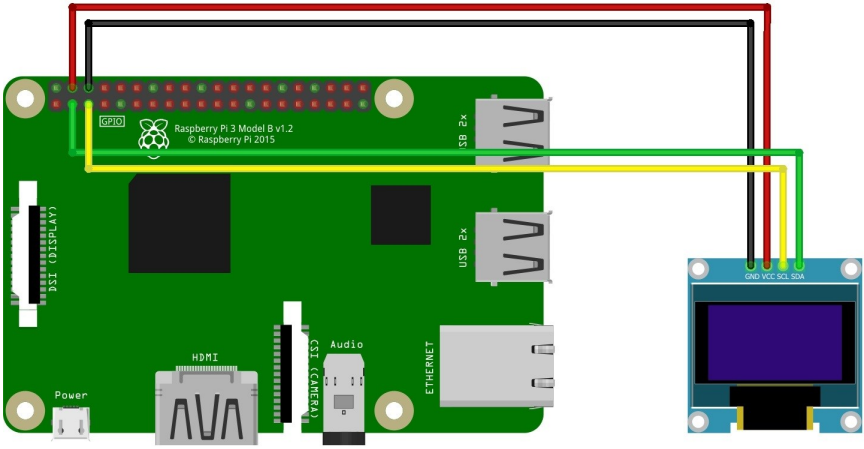
```
void setup()   {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.drawBitmap(0, 0, logo, 128, 64, WHITE);
  display.display();
}

void loop() {
}
```

2. USE WITH THE RASPBERRY PI

Connecting the display

Connect the display like shown in the picture or rather in the chart, with the pins of the Raspberry Pi.



SBC-OLED01	Raspberry Pi
GND	Pin 6 (Ground)
VCC	Pin 4 (5V) or Pin 1 (3.3V)
SCL	Pin 5 (BCM 3 / SCL)
SDA	Pin 3 (BCM 2 / SDA)

Mechanism of the display

As soon as the system has started, open the terminal console and perform the following commands:

```
sudo apt-get update
```

```
sudo apt-get install python-pip python-dev build-essential
```

```
sudo pip install RPi.GPIO
```

```
sudo apt-get install python-pil
```

```
sudo apt-get install python-smbus i2c-tools
```

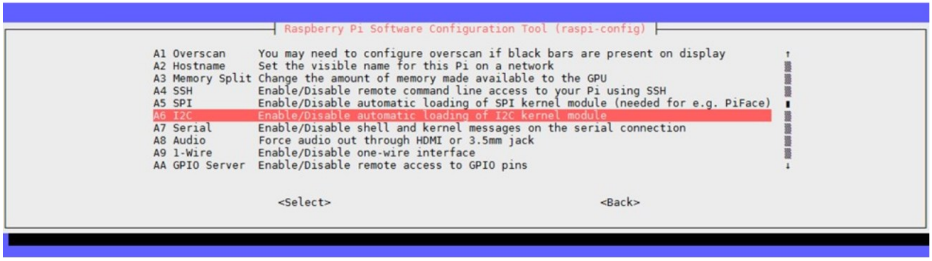
If the I2C function is still not activated on your Raspberry Pi,then you have to catch up on the settings.

For that, open and activate the settings with the following command:

```
sudo raspi-config
```

Select in the opening window the option **Interfacing Options**.

Now select and activate the option **I2C**.



Installation of the library

After the activation of the I2C function, we use the Python library with the following commands:

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

```
git clone https://github.com/adafruit/
Adafruit_Python_SSD1306.git
```

```
cd Adafruit_Python_SSD1306
```

```
sudo python setup.py install
```

Communication with the display

Now, you can begin with the first communication wit the display. For that, perform the following command:
(Notice that the display has to be connected for that)

```
sudo i2cdetect -y 1
```

The output will be the following:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:				--	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	--	3c	--	--	--
40:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70:	--	--	--	--	--	--	--									

The just started verification has recognized an I2C-signal at the address **3C**. This is the standard address of the display and is preselected in the Python library which we use.

Attention:

If the address of the I2C signal is located at **3D** , you must change this in the Python library. If the address is **3C** , you can proceed to the next point **code example**.

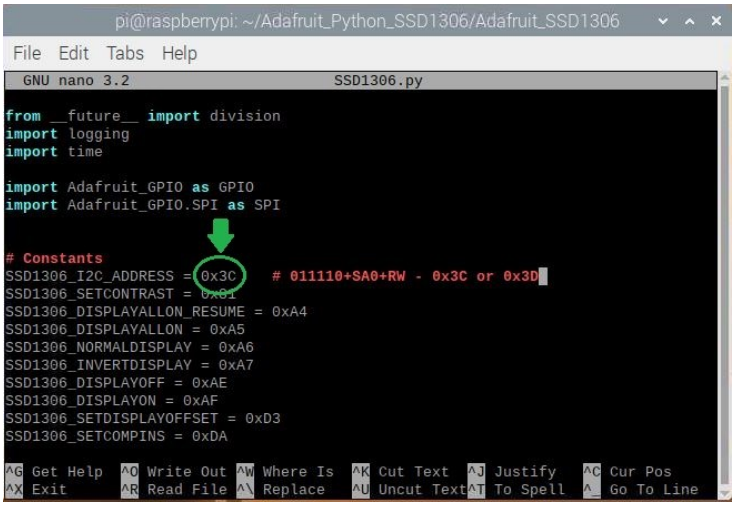
First of all, you must changeto the necessary subfolder with the following command:

```
cd Adafruit_SSD1306
```

Open the file which must be changed with the following command:

```
sudo nano SSD1306.py
```

Change now in the marked area the address from 3D to 3C.



Save and leave the file with the following key combination:
CTRL + O , ENTER , CTRL + X
Before you proceed with the manual leave the subfolder with the command:

```
cd ..
```

Code example

In the following, you can transfer the code example to use the display. We have to change in the folder of our library and create there a file with the following commands:

```
cd examples
```

```
sudo nano text.py
```

Copy this code completely into your open file:

Code example 1 (text)

```
import time
import Adafruit_GPIO.SPI as SPI
import Adafruit_SSD1306

from PIL import Image, ImageDraw, ImageFont

RST = 24

disp = Adafruit_SSD1306.SSD1306_128_64(rst=RST)

disp.begin()

disp.clear()
disp.display()
```

```
width = disp.width
height = disp.height
image = Image.new('1', (width, height))

draw = ImageDraw.Draw(image)

font = ImageFont.load_default()

x = 40
top = 20

draw.text((x, top),    'SBC-OLED01',
font=font, fill=255)
draw.text((x, top+20), 'Joy-IT', font=font,
fill=255)

disp.image(image)
disp.display()
time.sleep(.1)
```

Save and leave the file with the following key combination:
CTRL + O , ENTER , CTRL + X
The code example is therefore finished and can be operated with the following command:

```
sudo python text.py
```

3. OTHER INFORMATION

Our Information and Take-back Obligations according to the German Electronic Law (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 Handel 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.

4. SUPPORT

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

E-Mail: 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

Published: 02.07.2020