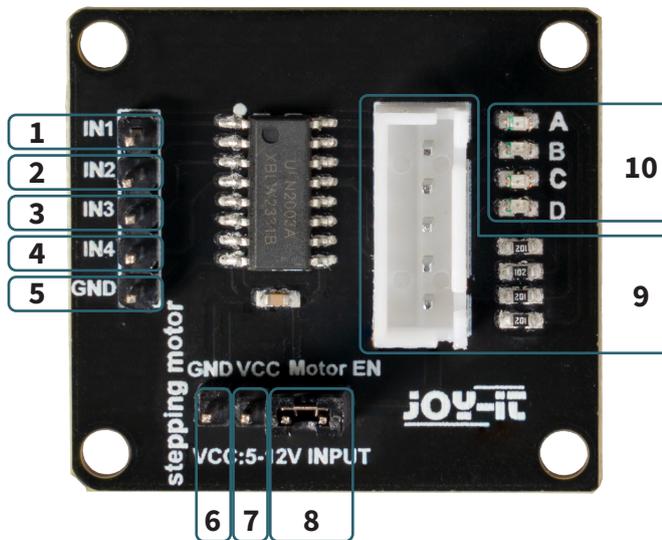# 28BYJ-48 STEPPER MOTOR

SBC-Moto1

JOY-IT

1. GENERAL INFORMATION

Dear customer,

thank you for choosing our product. Below we will show you what you need to bear in mind during commissioning and use.

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

2. DEVICE OVERVIEW

| DIGIT	FUNCTION
1	IN1; control input for motor connection A
2	IN2; control input for motor connection B
3	IN3; control input for motor connection C
4	IN4; control input for motor connection D
5	GND; ground connection supply voltage
6	GND; ground connection supply voltage
7	VCC; supply voltage 5-12 V; note the required supply voltage of the motor you are using
8	Enable; remove the jumper to cut the power supply to the motor
9	Motor connection
10	Status LEDs for the individual motor connections (active-LOW)

3. ARDUINO APPLICATION EXAMPLE

First connect the module to your Arduino:
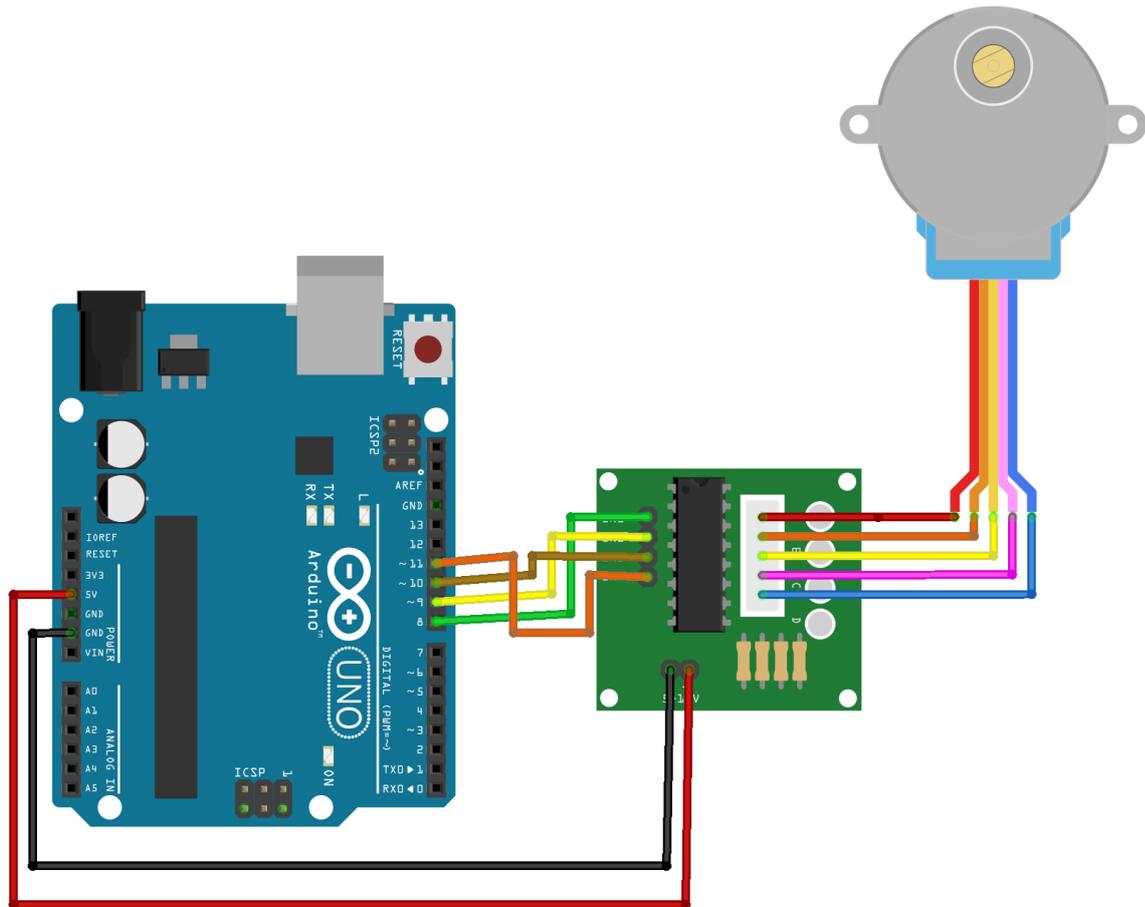
SBC-MOTO1	ARDUINO
GND	GND
VCC	5 V
IN1	D8
IN2	D9
IN3	D10
IN4	D11

Before you transfer the source code below to your Arduino, the **CheapStepper** library must first be added. This library extends your Arduino library with the required range of functions and ensures that the product can be used correctly. This library was released by Tyler Henry under the GNU General Public License and is available <u>here</u> and can be viewed here.

first open your Arduino IDE and select the library management as shown in the following image.
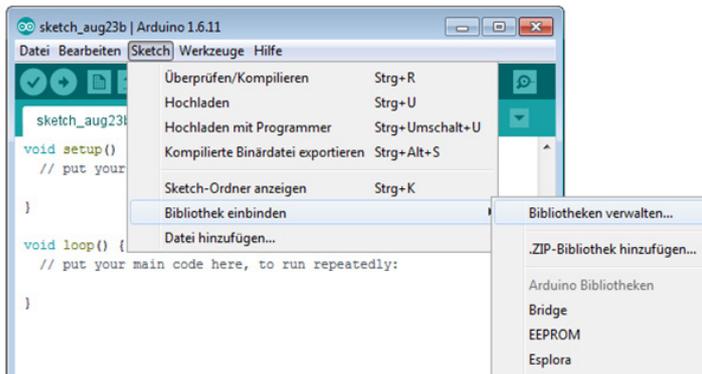
In the library manager that now opens, you will find the CheapStepper library under the search term **28BYJ-48**. Install it.
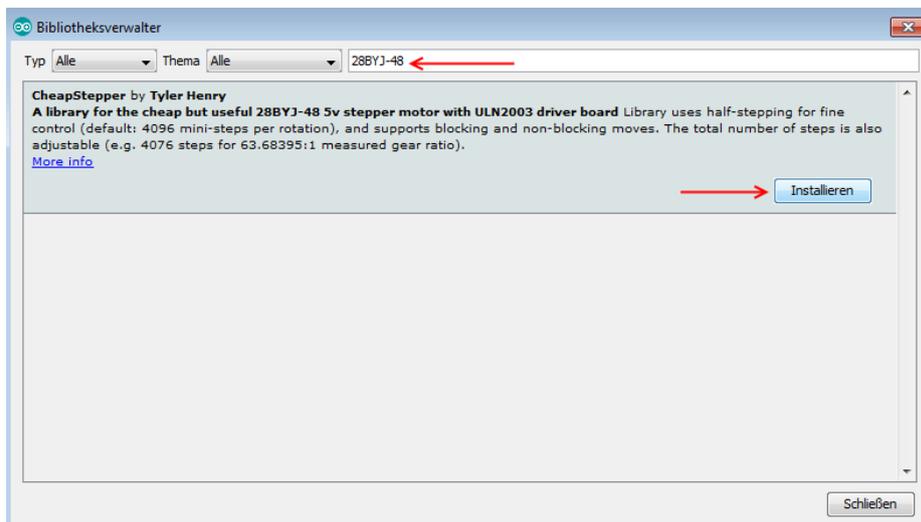
With the following code example, you can use the motor with an Arduino. In this example, the motor rotates repeatedly first one full revolution in one direction and then one full revolution in the other direction. The example can be adapted to your requirements, depending on the intended use.

```cpp
#include <CheapStepper.h>
CheapStepper stepper;

//Direction assignment of the motor
boolean moveClockwise = true;

void setup()
{
  Serial.begin(9600);
  Serial.println("28BYJ-48 ready.");
}

void loop()
{
 for(int s=0; s<4096; s++){
    //One full rotation contains 4096 steps
    stepper.step(moveClockwise);
    //Output of the current motor position in the console
   int nStep = stepper.getStep();
   if (nStep%64==0)
    {
      Serial.print("current step position: "); Serial.print(nStep);
      Serial.println();
    }
  }
  delay(1000);
  //Change of direction after complete rotation
  moveClockwise = !moveClockwise;
}
```
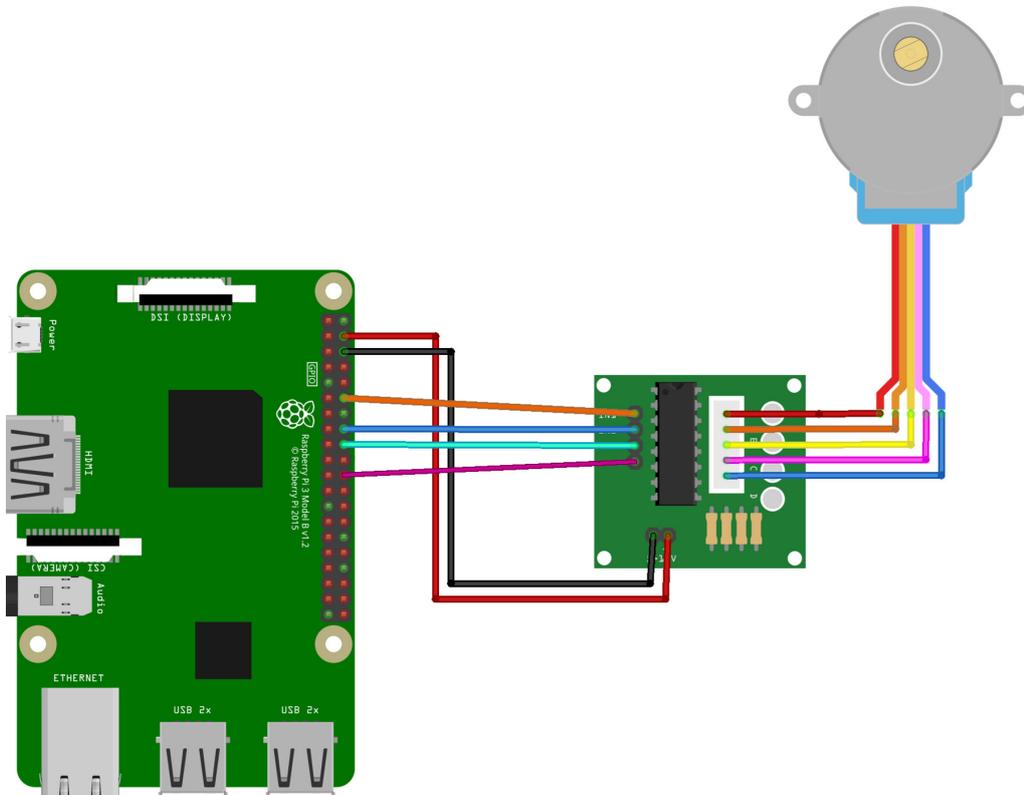
> **i** These instructions were written under Raspberry Pi OS Bookworm for the Raspberry Pi 4 and 5. It has not been checked with newer operating systems or hardware.

First connect the module to your Raspberry Pi.

SBC-MOTO1	RASPBERRY PI
GND	GND
VCC	5V
IN1	GPIO18
IN2	GPIO23
IN3	GPIO24
IN4	GPIO25

You can now download the code example we have prepared here or with the following command:

```
wget https://joy-it.net/files/files/Produkte/SBC-Moto1/SBC-Moto1-RPi.zip
```

Now you can unpack the file with the following command:

```
unzip SBC-Moto1-RPi.zip
```

You can now start the example with this command:

```
python3 SBC-Moto1-RPi.py
```

Alternatively, you can also manually copy the code below into your Python file

```python
from time import sleep
from gpiozero import OutputDevice
# Pin assignment on the Raspberry Pi
A = OutputDevice(18)
B = OutputDevice(23)
C = OutputDevice(24)
D = OutputDevice(25)
delay_time = 0.001   # 1 millisecond
# Driving the coils of the motor
def step1():
    D.on()
    sleep(delay_time)
    D.off()
def step2():
    D.on()
    C.on()
    sleep(delay_time)
    D.off()
    C.off()
def step3():
    C.on()
    sleep(delay_time)
    C.off()
def step4():
    B.on()
    C.on()
    sleep(delay_time)
    B.off()
    C.off()
def step5():
    B.on()
    sleep(delay_time)
    B.off()
def step6():
    A.on()
    B.on()
    sleep(delay_time)
    A.off()
    B.off()
def step7():
    A.on()
    sleep(delay_time)
    A.off()

def step8():
    D.on()
    A.on()
    sleep(delay_time)
    D.off()
    A.off()

# Perform one complete rotation
for _ in range(512):
    step1()
    step2()
    step3()
    step4()
    step5()
    step6()
    step7()
    step8()
```

Our information and take-back obligations under the German Electrical and Electronic Equipment Act (ElektroG)

Symbol on electrical and electronic equipment:
This crossed-out garbage can means that electrical and electronic appliances **do not** belong in household waste. You must hand in the old appliances at a collection point. Before handing them in, you must separate used batteries and accumulators that are not enclosed by the old appliance.

Return options:
As an end user, you can hand in your old appliance (which essentially fulfills the same function as the new appliance purchased from us) for disposal free of charge when purchasing a new appliance. Small appliances with no external dimensions greater than 25 cm can be disposed of in normal household quantities regardless of whether you have purchased a new appliance.

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

Return option in your area:
We will send you a parcel stamp with which you can return the device to us free of charge. To do so, please contact us by e-mail at Service@joy-it.net or by telephone.

Packaging information:
Please pack your old appliance securely for transportation. 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

We are also there for you after your purchase. If you still have any questions or problems arise, we are also available by e-mail, telephone and ticket support system.

E-Mail: service@joy-it.net
Ticket-System: https://support.joy-it.net
Phone: +49 (0)2845 9360 - 50 (Mon - Thur: 09:00 - 17:00 o'clock CET,
                                      Fri: 09:00 - 14:30 o'clock CET)

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

Published: 2024.05.08