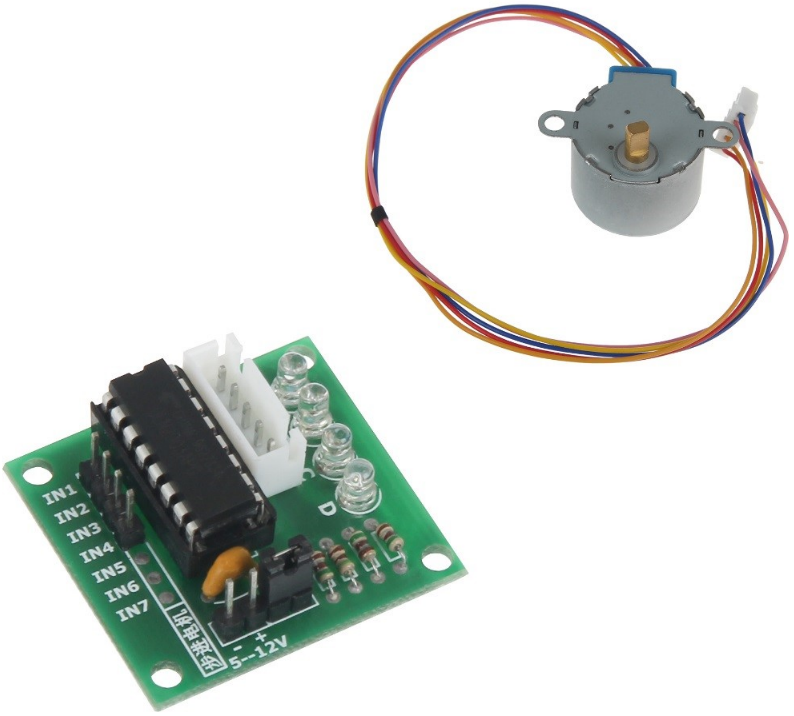
Moto1

28BYJ-48 Stepper Motor



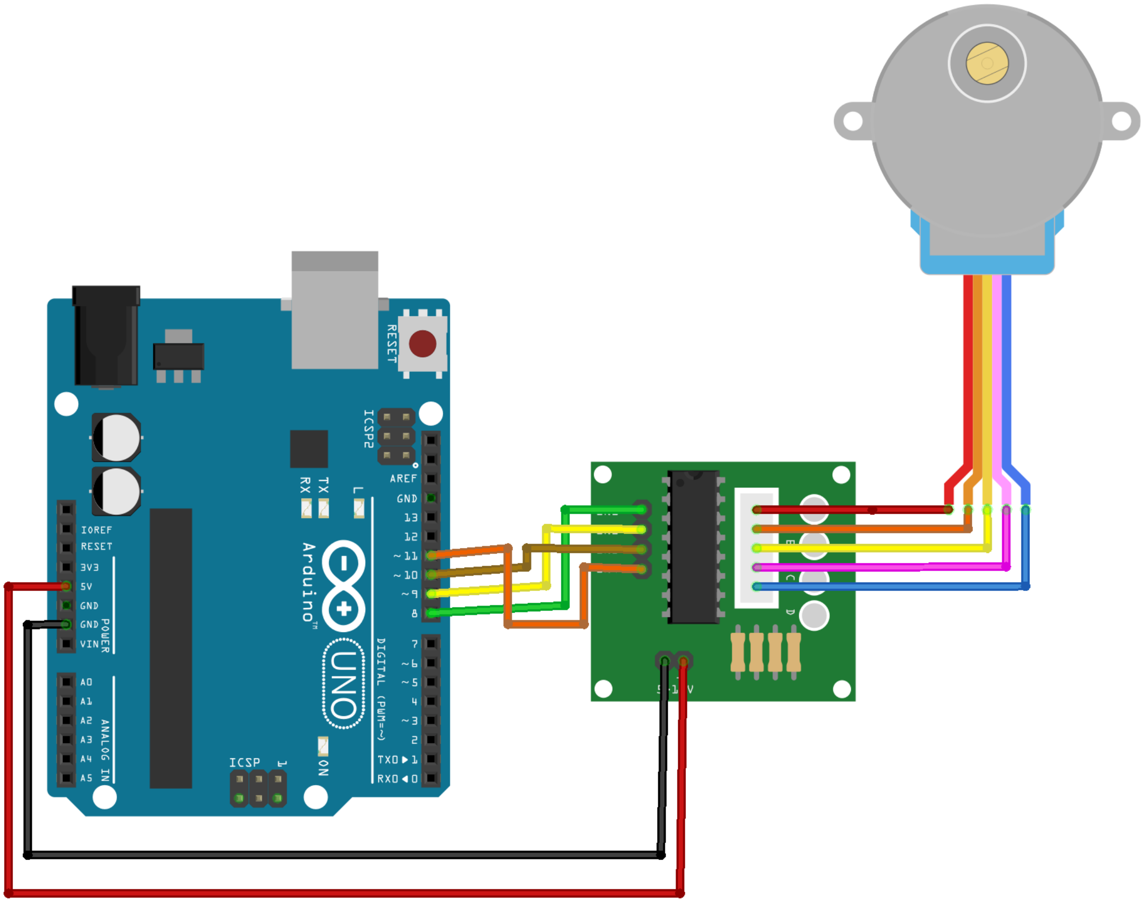
Index

Dear customer,
thank you for purchasing our product.
Please find our instructions below.

1. Using with an Arduino

1.1 Connecting the motor

Connect the motor to the driver-board with the pin connector.
Connect the board, as seen in the following image and in the following table, to the pins of your Arduino.

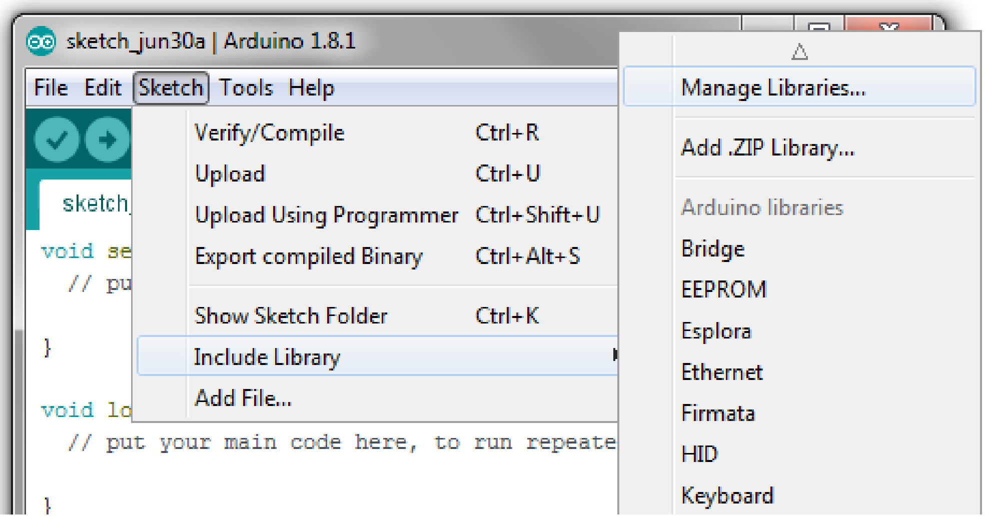


Arduino PIN	Driver-Board PIN
+5V	+
GND	-
8	INT1
9	INT2
10	INT3
11	INT4

1.2 Installing the library

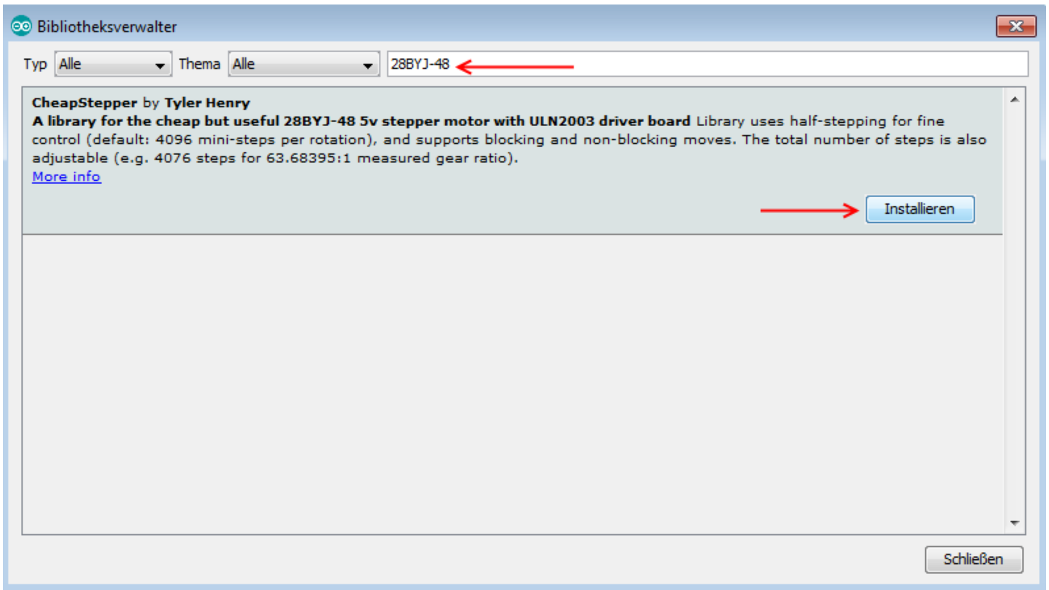
Before transferring the code to your Arduino, you need to add the **CheapStepper** library.
This library extends your Arduino library by the needed functions to provide a perfect usage of the product.
This library was made by **Tyler Henry** and was published under the **GNU General Public License** and can be accesed here.

Open your Arduino DIE and select, as seen in the following image, the Arduino library manager.



```
void se
  // pu
}

void lo
  // put your main code here, to run repeate
}
```

Now you can install the **CheapStepper** library by searching for **28BYJ-**48.



CheapStepper by **Tyler Henry**
A library for the cheap but useful 28BYJ-48 5v stepper motor with ULN2003 driver board Library uses half-stepping for fine control (default: 4096 mini-steps per rotation), and supports blocking and non-blocking moves. The total number of steps is also adjustable (e.g. 4076 steps for 63.68395:1 measured gear ratio).
More info

1.3 Using the motor

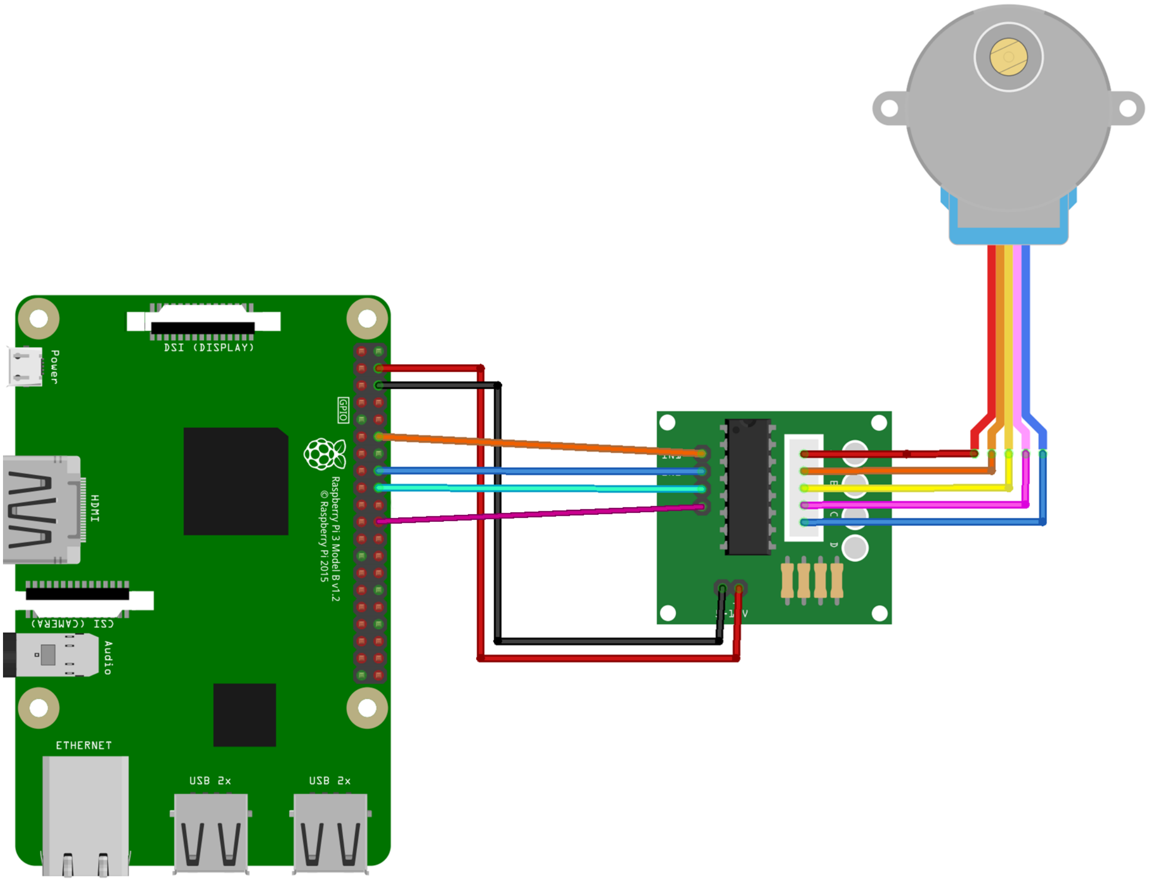
With the following example, you can use the motor with an Arduino.
In this specific example, the motor rotates in to the one direction and then to the other.
You can, of course, modify this example to fit your needs.

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

//Direction setup
boolean moveClockwise = true;

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

void loop()
{
  for (int s=0; s<4096; s++){
    //one complete turn contains 4096 steps
    stepper.step(moveClockwise);
    //printing the current position to the console
    int nStep = stepper.getStep();
    if (nStep%64==0)
    {
      Serial.print("current step position: "); Serial.print(nStep);
      Serial.println();
    }
  }
  delay(1000);
  //change direction after one turn
  moveClockwise = !moveClockwise;
}
```

1. Using with a Raspberry Pi

1.1 Connecting the motor

Connect the motor to the driver-board with the pin connector.
Connect the board, as seen in the following image and in the following table, to the pins of your Raspberry Pi.



Raspberry PIN	Driver-Board PIN
PIN 4 (5V)	+
PIN 6 (Ground)	-
PIN 12 (BCM 18)	INT1
PIN 16 (BCM 23)	INT2
PIN 18 (BCM 24)	INT3
PIN 22 (BCM 25)	INT4

2.2 Installing the software

You can skip this step and continue with Step 2.3 if you are already using the latest Raspbian system on your Raspberry.

Install the latest Raspbain Image to your SD-Card with the„**Win32 Disk Imager**“-Tool.
You can find the latest Raspbian image here.

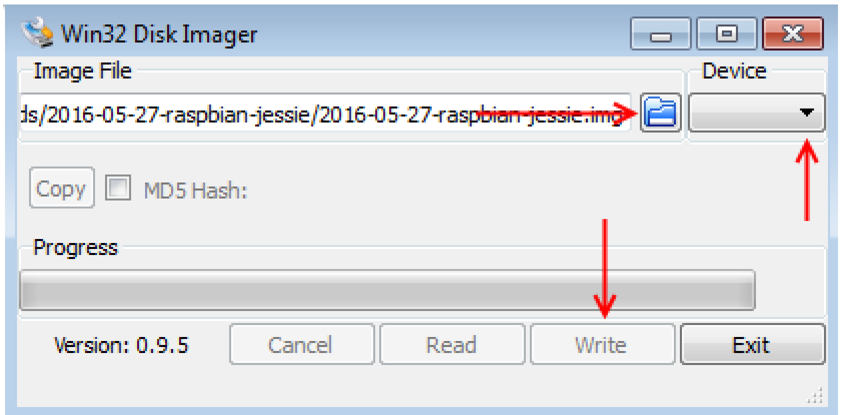


Image File: ds/2016-05-27-raspbian-jessie/2016-05-27-raspbian-jessie.img

2.3 Using the motor

You can use the motor with the following code example.
In this example, the motor will make one complete turn.
To create this example, open the terminal and create a new file:

```
sudo nano motor.py
```

Enter the following code in the now opened editor.
You can save the file by pressing **CTRL+O** and quit by pressing **CTRL+X**.

```
from time import sleep
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

# PIN-Assignment
A=18
B=23
C=24
D=25
time = 0.001

# defining the PINs
GPIO.setup(A,GPIO.OUT)
GPIO.setup(B,GPIO.OUT)
GPIO.setup(C,GPIO.OUT)
GPIO.setup(D,GPIO.OUT)
GPIO.output(A, False)
GPIO.output(B, False)
GPIO.output(C, False)
GPIO.output(D, False)

# driving the motor
def Step1():
    GPIO.output(D, True)
    sleep (time)
   GPIO.output(D, False)
```

```python
def Step2():
    GPIO.output(D, True)
    GPIO.output(C, True)
    sleep (time)
    GPIO.output(D, False)
    GPIO.output(C, False)

def Step3():
    GPIO.output(C, True)
    sleep (time)
    GPIO.output(C, False)

def Step4():
    GPIO.output(B, True)
    GPIO.output(C, True)
    sleep (time)
    GPIO.output(B, False)
    GPIO.output(C, False)

def Step5():
    GPIO.output(B, True)
    sleep (time)
    GPIO.output(B, False)

def Step6():
    GPIO.output(A, True)
    GPIO.output(B, True)
    sleep (time)
    GPIO.output(A, False)
    GPIO.output(B, False)

def Step7():
    GPIO.output(A, True)
    sleep (time)
    GPIO.output(A, False)

def Step8():
    GPIO.output(D, True)
    GPIO.output(A, True)
    sleep (time)
    GPIO.output(D, False)
    GPIO.output(A, False)
```

```
# start one complete turn
for i in range (512):
    Step1()
    Step2()
    Step3()
    Step4()
    Step5()
    Step6()
    Step7()
    Step8()

GPIO.cleanup()
```

After you saved your code and quit the edior, you can start the example by entering:

```
sudo python motor.py
```

3. Support

We also support you after your purchase. If you have any questions left or encounter any problems, feel free to contact us by E-Mail, phone or with our ticket-supportsystem on our website.

E-Mail: service@joy-it.net

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

Phone: +49 (0)2845 98469 – 66 (11- 18 Uhr)

For further informations, please visit our website:

www.joy-it.net