Ein Bild, das Text, Elektronik, Screenshot, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

<https://www.youtube.com/watch?v=Vtbd80FksuM>

Ciao Walti

Anbei das Datasheet und der Anschlussplan. Dieser beinhaltet noch einen Fehler und zwar GPIO 25 ist auf dem Pin 22 und nicht wie im Schaltplan auf Pin20.

Im Code habe ich mal auf dem Beispiel des Herstellers aufgebaut.

Gruss, Ruedi

**Ruedi Kubli**  
CISSP, CISA, FCP, FCSS (SASE, OT)  
Business Development Manager OT/IoT

|  |  |  |
| --- | --- | --- |
|  |  | Boll Engineering AG | Jurastrasse 58 | 5430 Wettingen | Switzerland Main: +41 56 437 60 60 | Direct: +41 56 437 67 13 | Mobile: +41 79 655 88 88 [rku@boll.ch](mailto:%20rku@boll.ch) | [www.boll.ch](https://www.boll.ch/) | [blog.boll.ch](https://blog.boll.ch/) | [vcard](https://www.boll.ch/contact/195513628085068) |

Ein Bild, das Elektronik, Elektronisches Bauteil, Elektrisches Bauelement, Schaltung enthält.

KI-generierte Inhalte können fehlerhaft sein.

# DC 5V Stepper Motor 28BYJ-48

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produktentschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:

# Verwendung mit einem Arduino

## Schritt 1 – Anschließen des Motors

**Ein Bild, das Kreis, Kabel, Text enthält.

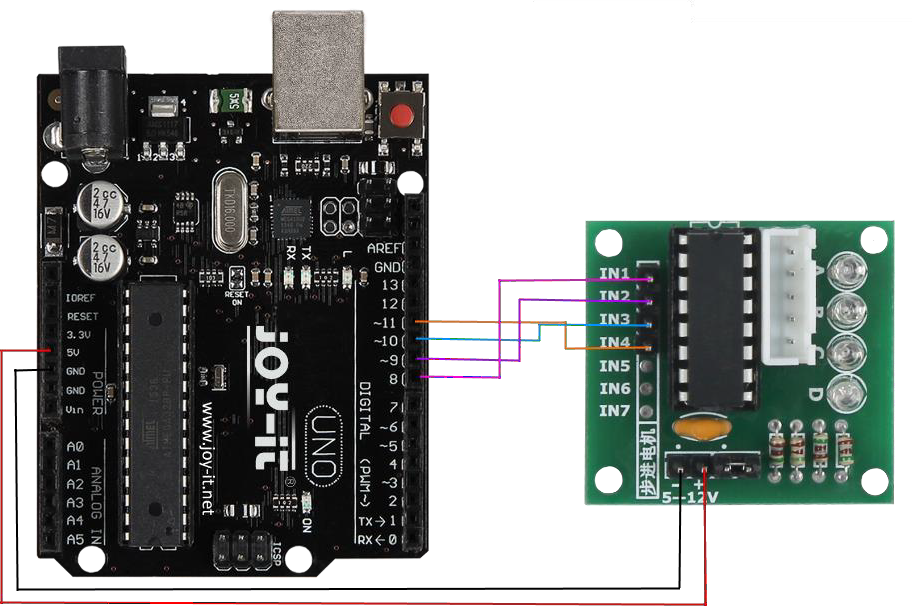
KI-generierte Inhalte können fehlerhaft sein.Ein Bild, das Elektronik, Schaltung, Elektronisches Bauteil, Elektrisches Bauelement enthält.

KI-generierte Inhalte können fehlerhaft sein.**

**Bild 1: 28BYJ-48 mit Treiber-Board**

Schließen Sie den Schrittmotor an den auf dem Treiberboard vorgesehen Anschluss an.

Schließen Sie das das Board, wie im folgenden Bild 2, bzw. in der folgenden Tabelle 1, zu sehen, an die PINs des Arduinos an.



**Bild 2: Verkabelung mit Arduino**

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

**Tabelle 1: PIN-Verbindung zwischen Arduino und Treiber-Board**

## Schritt 2 – Installation der Bibliothek

Bevor Sie den unten befindlichen Quellcode auf Ihren Arduino übertragen, muss zunächst die

**CheapStepper** Bibliothek hinzugefügt werden.

Dazu klicken Sie bitte, wie in Bild 3 zu sehen, auf Sketch  Bibliothek einbinden  Bibliotheken verwalten.

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Bild 3: Bibliothekverwaltung im Arduino-Sketch**

Im sich darauf öffnenden Bibliotheksverwalter suchen Sie unter dem Suchbegriff **28BYJ-48** die

**CheapStepper** Bibliothek und installieren diese.

Ein Bild, das Text, Screenshot, Software, Display enthält.

KI-generierte Inhalte können fehlerhaft sein.

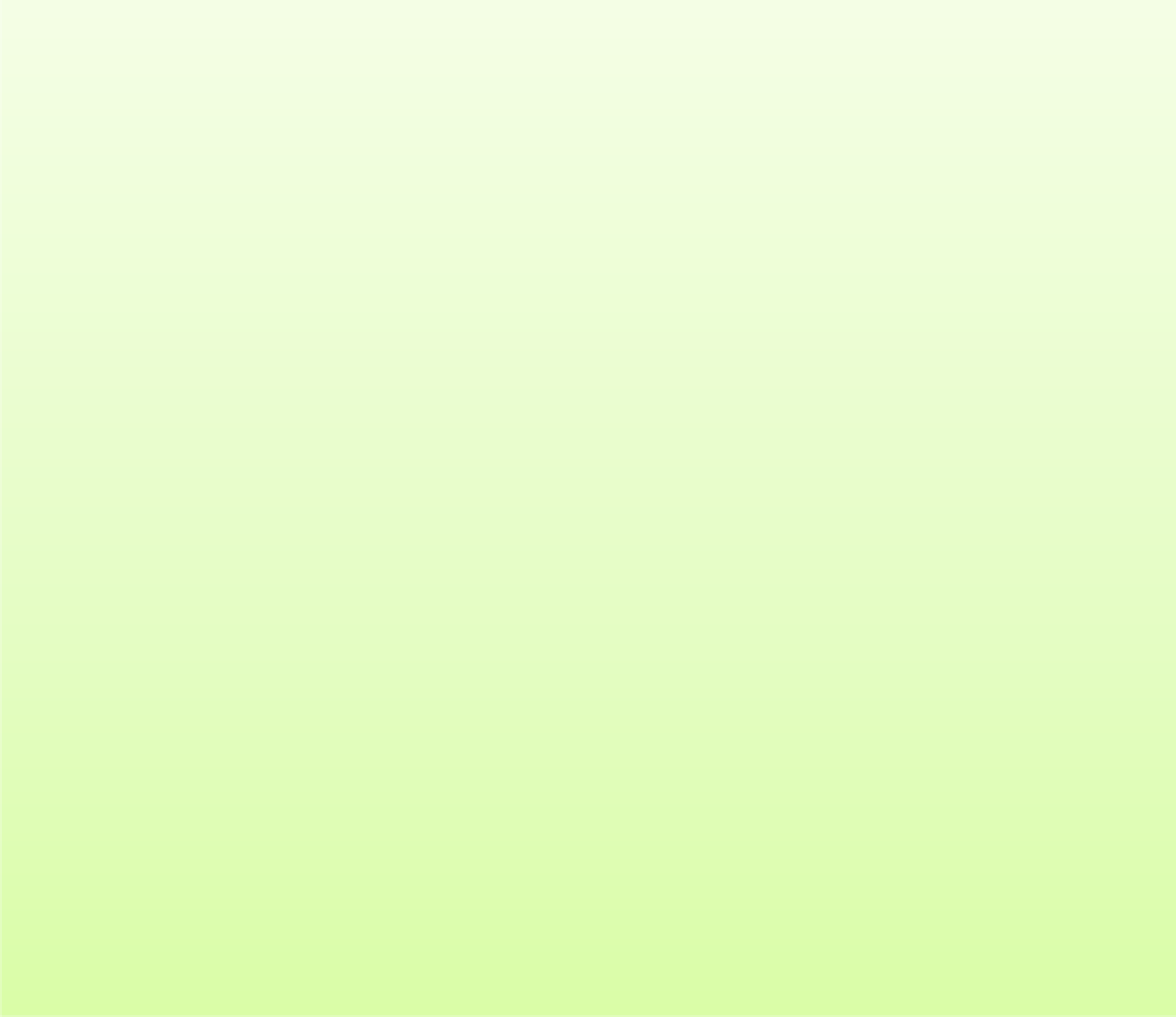
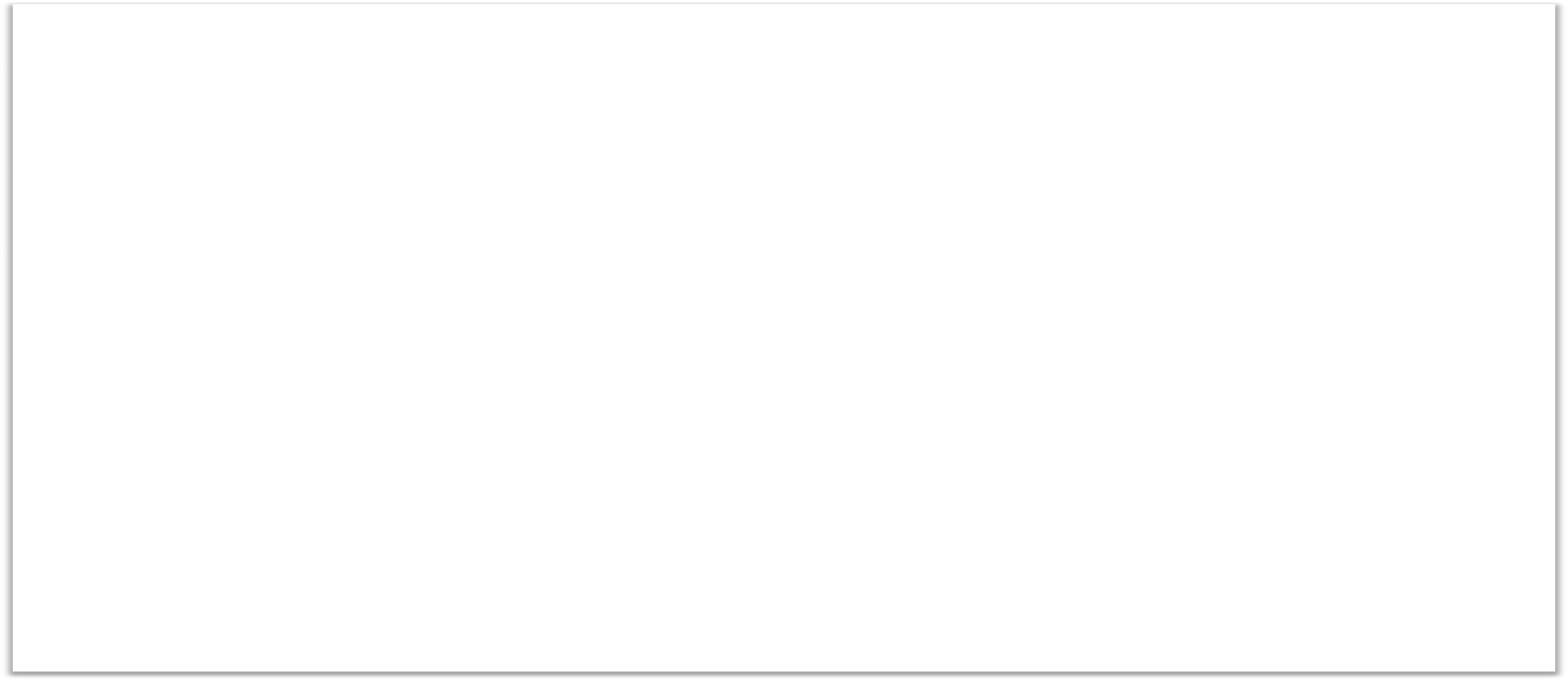
**Bild 4: Installation der CheapStepper Bibliothek**

## Schritt 3 – Installation des Motors

Mit folgendem Codebeispiel, können Sie den Motor mit einem Arduino verwenden.

In diesem Beispiel rotiert der Motor wiederholend zuerst eine volle Umdrehung in die eine Richtung und danach eine volle Umdrehung in die andere Richtung.

Das Beispiel kann, je nach Verwendungszweck, von Ihnen an ihre Wünsche angepasst werden.



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

//Richtungszuweisung des Motors boolean moveClockwise **=** true**;**

void setup**()**

**{**

Serial**.**begin**(**9600**);** Serial**.**println**(**"28BYJ-48 bereit."**);**

**}**

void loop**()**

**{**

**for (**int s**=**0**;** s**<**4096**;** s**++){**

//Eine volle Umdrehung beinhaltet 4096 Schritte stepper**.**step**(**moveClockwise**);**

//Ausgabe der aktuellen Motorposition in der Konsole int nStep **=** stepper**.**getStep**();**

**if (**nStep**%**64**==**0**)**

**{**

Serial**.**print**(**"current step position: "**);** Serial**.**print**(**nStep**);** Serial**.**println**();**

**}**

**}**

delay**(**1000**);**

//Richtungswechsel nach vollständiger Umdrehung moveClockwise **= !**moveClockwise**;**

**}**

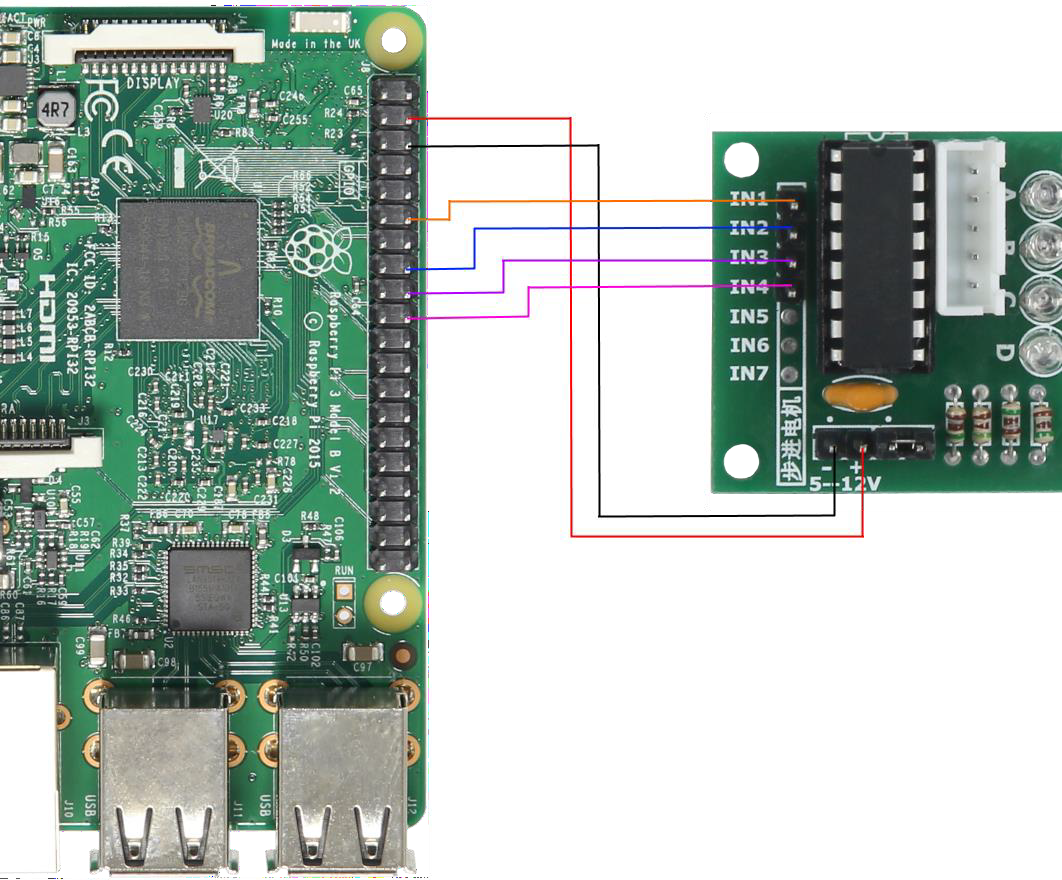
**Code 1: Arduino Quellcode**

# Verwendung mit einem Raspberry Pi

## Schritt 1 – Anschließen des Motors

Schließen Sie den Schrittmotor an den auf dem Treiberboard vorgesehen Anschluss an.

Schließen Sie das das Board, wie im folgenden Bild 5, bzw. in der folgenden Tabelle 2, zu sehen, an die PINs des Raspberry Pis an.



**Bild 5: Verkabelung mit Raspberry Pi**

|  |  |
| --- | --- |
| **Raspberry Pi PIN** | **Treiber-Board PIN** |
| PIN 12 (BCM 18) | IN1 |
| PIN 16 (BCM 23) | IN2 |
| PIN 18 (BCM 24) | IN3 |
| PIN 22 (BCM 25) | IN4 |
| PIN 4 (5v Power) | + |
| PIN 6 (Ground) | - |

**Tabelle 2: PIN-Verbindung zwischen Rasperry Pi und Treiber-Board**

## Schritt 2 – Installation der Software

Sollten Sie bereits ein aktuelles Raspbian-System auf Ihrem Raspberry verwenden, so können Sie diesen Schritt überspringen und sofort mit Schritt 3 fortfahren.

Installieren Sie auf Ihre SD-Karte mit Hilfe des [„Win32 Disk Imager“-](https://sourceforge.net/projects/win32diskimager/)Programms das aktuelle Raspbian Image, welches Sie unter dem folgenden [Link](https://downloads.raspberrypi.org/raspbian_latest) zum Download finden.

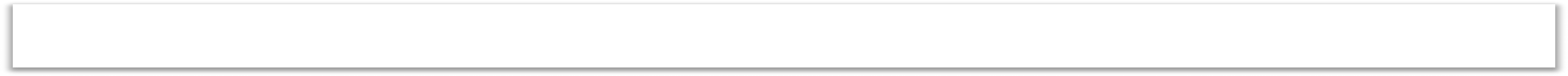
Ein Bild, das Text, Screenshot, Display, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Bild 6: Screenshot des Win32 Disk Imagers**

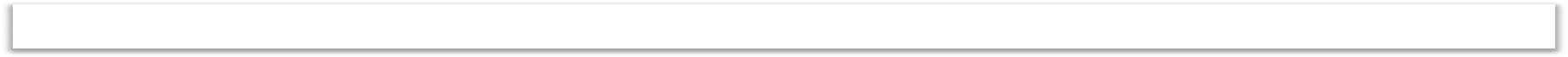
## Schritt 3 – Aktivierung des SPI-Moduls

Sobald Sie die Installation abgeschlossen und das System gestartet haben, öffnen Sie die Terminal- Konsole und führen Sie folgendes Kommando aus:



sudo apt-get install python-pip python-dev build-essential sudo pip install RPi.GPIO

**Code 2: Installation der GPIO Bibliothek**



sudo apt-get install python-imaging

**Code 3: Installation der Python Bibliothek**

## Schritt 4 – Installation des Motors

Mit folgendem Codebeispiel, können Sie den Motor mit einem Raspberry Pi verwenden. In diesem Beispiel rotiert der Motor eine vollständige Umdrehung.

Das Beispiel kann, je nach Verwendungszweck, von Ihnen an ihre Wünsche angepasst werden. Öffnen Sie zum Erstellen des Codebeispiels das Terminal und erstellen Sie eine neue Datei:

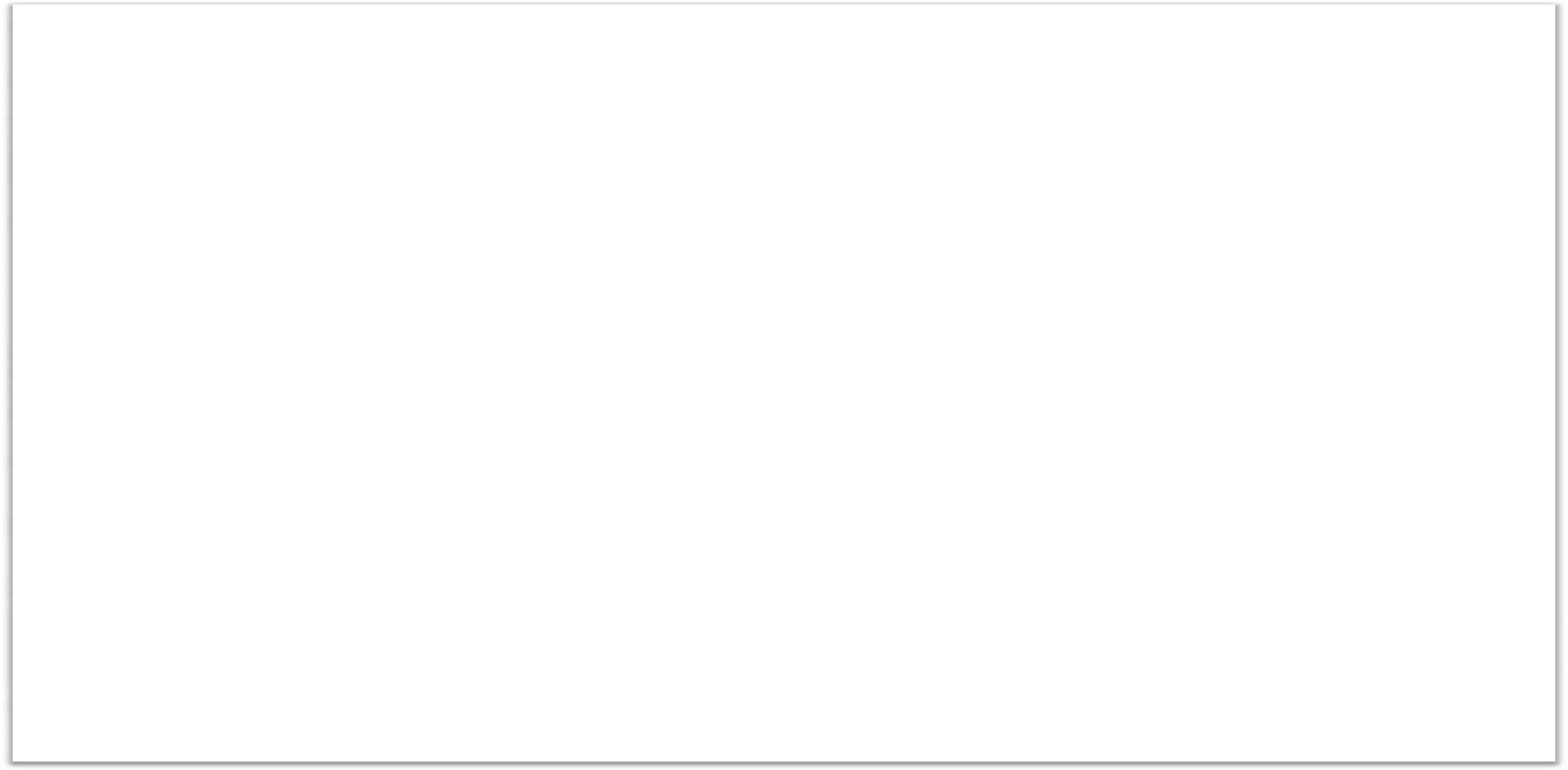


sudo nano motor.py

**Code 3: Quellcode-Datei erstellen**

Geben Sie den nun geöffneten Editor folgende Quellcode Teile.

Sie können Ihre Datei mit **Strg+O** speichern und mit **Strg+X** verlassen.



**from** time **import** sleep

**import** RPi**.**GPIO **as** GPIO GPIO**.**setmode**(**GPIO**.**BCM**)**

# PIN-Zuweisung am Raspberry A**=**18

B**=**23 C**=**24 D**=**25

time **=** 0.001

# PINS definieren 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)**

# Ansteuerung der Spulen des Motors

**def** Step1**():** GPIO**.**output**(**D**, True)** sleep **(**time**)** GPIO**.**output**(**D**, False)**

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

**Code 4: Teil 1 des Quellcodes**

**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)**

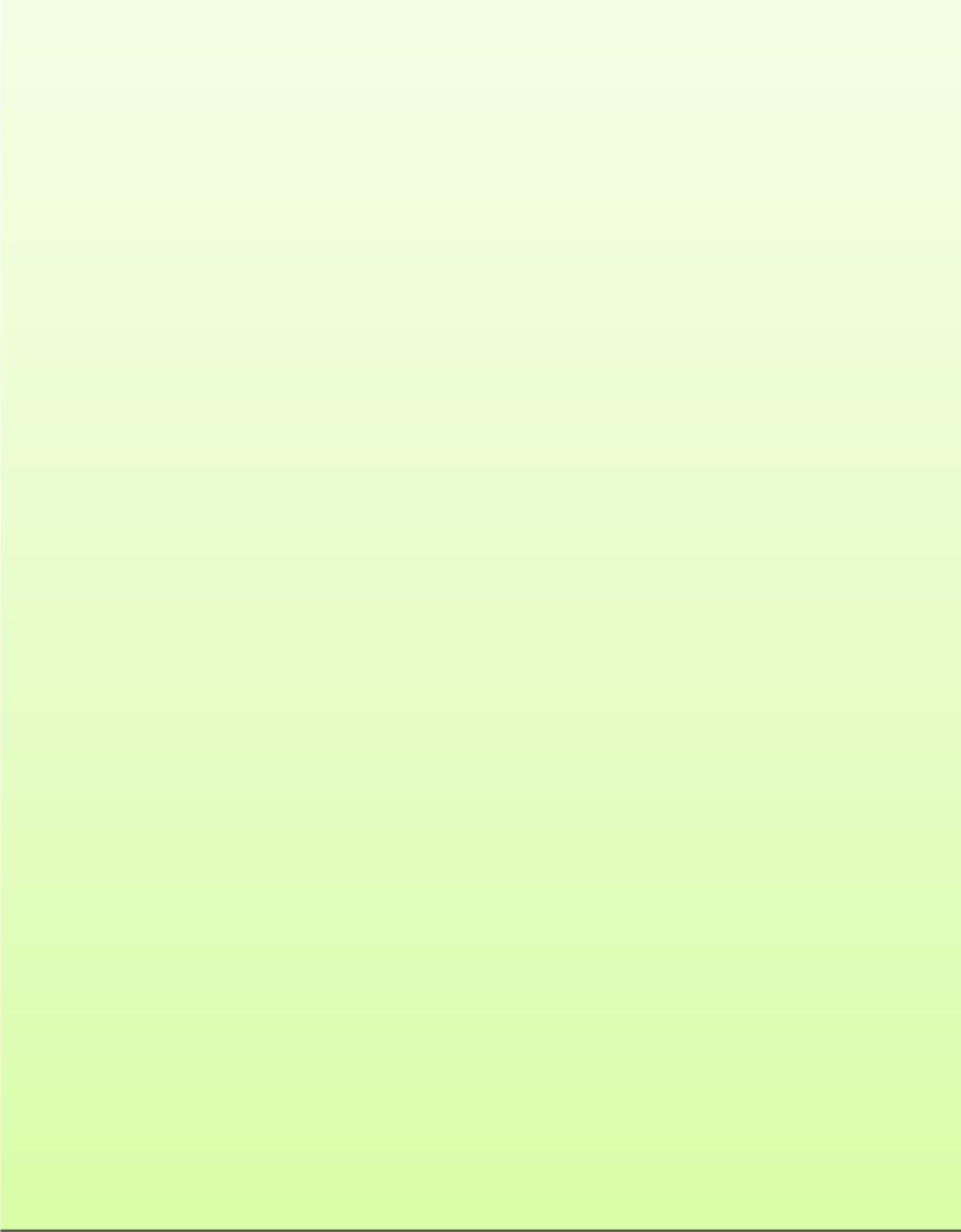
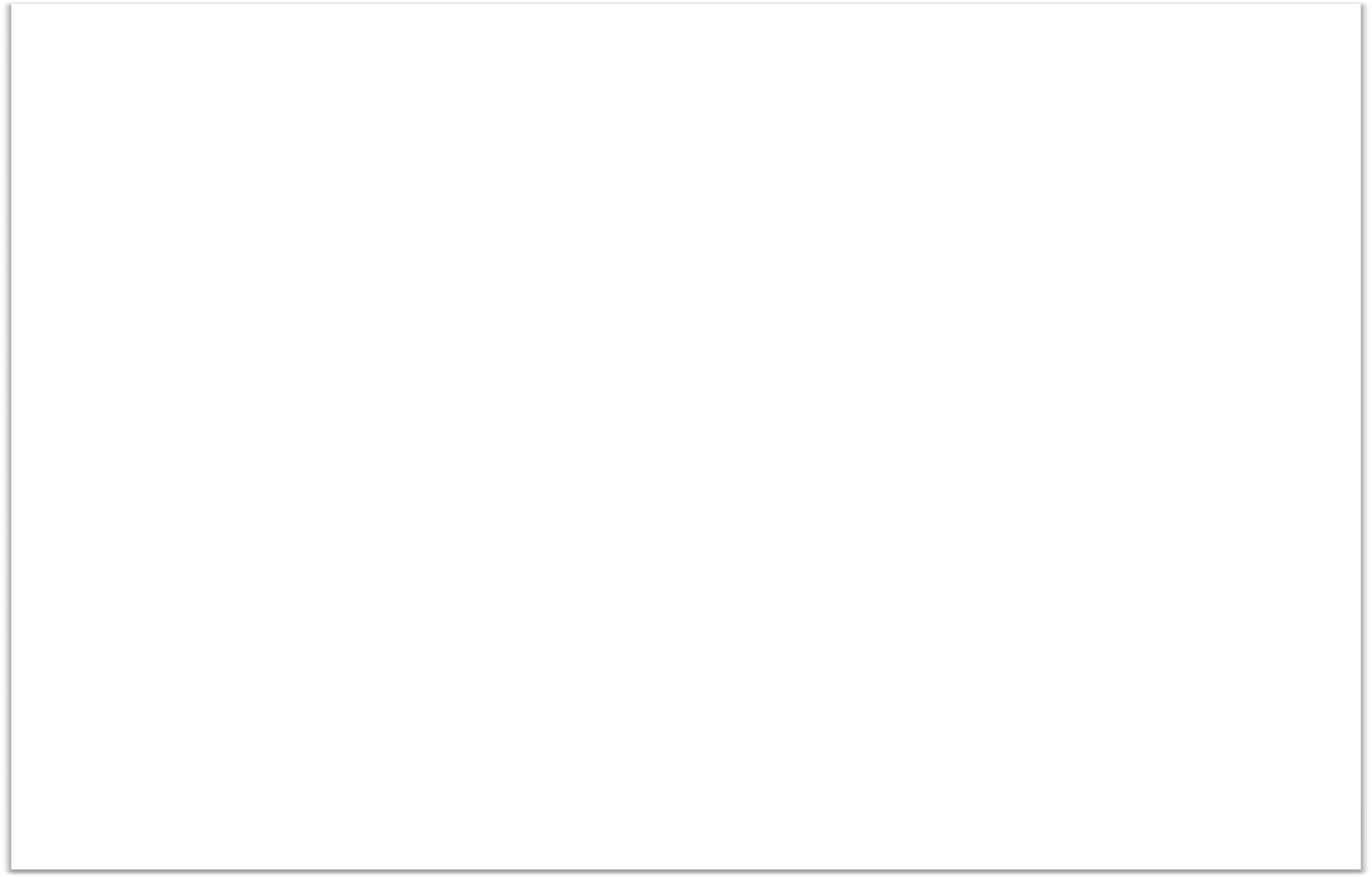
# Eine komplette Umdrehung starten

**for** i **in** range **(**512**):**

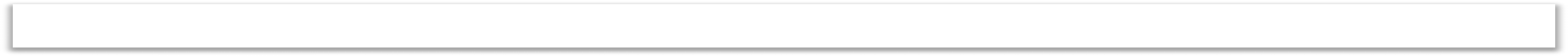
Step1**()** Step2**()** Step3**()** Step4**()** Step5**()** Step6**()** Step7**()** Step8**()**

GPIO**.**cleanup**()**

**Code 5: Teil 2 des Quellcodes**



Nachdem Sie den Quellcode gespeichert und den Editor verlassen haben, können Sie Ihren Code mit folgendem Befehl starten:



sudo python motor.py

**Code 6: Ausführen des Motor-Codes**