

BN 2134135

Schrittmotor-Treiber-Modul

DE Kurzanleitung

Version: 2

Lieferumfang

- Produkt

Beschreibung

Der Doppel-H-Treiber ist ein Hochspannungs-/Hochstrom-Doppel-Vollbrückentreiber mit einem L298N, einer integrierten monolithischen Schaltung in einem 15-Leiter-Multiwatt- und PowerSO20-Paket.

Es akzeptiert Standard-TTL-Logic-Levels und steuert induktive Lasten wie Relais, Magnete, Gleichstrom- und Schrittmotoren.

Jeder Treiber verfügt über einen Freischalt-Eingang, der es Ihnen ermöglicht, den Treiber unabhängig von den Eingangssignalen zu aktivieren/deaktivieren.

Die Emitter der unteren Transistoren jeder Brücke sind miteinander verbunden.

Verwenden Sie die passenden externen Anschlüsse, um einen externen Messwiderstand anzuschließen.

Die zusätzliche Spannungsversorgung stellt sicher, dass die Logik mit einer niedrigeren Spannung arbeiten kann.

Entwickler Ressourcen

Entwickler-Ressourcen und zusätzliche Produktinformationen sind verfügbar über:



<https://www.conrad.de/>

Pinout / Pin Map

Pin Name	Beschreibung
MOTORA	Motor 1
MOTORB	Motor 2
VMS	5 V/DC - 35 V/DC
GND	Masse
5V	Stromeingang für den Logikschaltkreis auf der Platine

Pin Name	Beschreibung
ENA	Freigabe Motor 1
IN1	Steueranschluss Motor 1
IN2	Steueranschluss Motor 1
ENB	Freigabe Motor 2
IN3	Steueranschluss Motor 2
IN4	Steueranschluss Motor 2
5V	5V Ausgang
GND	Masse
CSB	Stromprüfanschluss für Motor 2: Widerstand zur Stromprüfung anschliessen oder mit Steckbrücke sperren.
CSA	Stromprüfanschluss für Motor 1: Widerstand zur Stromprüfung anschliessen oder mit Steckbrücke sperren.
UR1	Pullup-Widerstand
UR2	Pullup-Widerstand
UR3	Pullup-Widerstand
UR4	Pullup-Widerstand
5V_EN	5V Steckbrücke (Source): Mit Steckbrücke: 5V-Versorgung der Logikschaltung über VMS-Port ($7V < VMS < 18V$) Steckbrücke entfernt: 5V-Versorgung der Logikschaltung über 5V-Port

Anwendungsbeispiel - 1

Dieses Beispiel zeigt die Verwendung eines 2-Kanal-DC-Motors.

Die Anweisungen basieren auf der Arduino®-Plattform. Sie können auch ein Arduino Derivat oder eine andere Plattform verwenden, die das Produkt unterstützt.

Anschluss

Modul	IN1	IN2	IN3	IN4
Arduino®	13	12	11	10

Code

```
int in1=13;
int in2=12;
int in3=11;
int in4=10;
```

```

int speedPinA=6;
int speedPinB=5;
void setup() {
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  digitalWrite(in1,HIGH);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,HIGH);
}
void loop() {
  _mRight(in1,in2);
  _mRight(in3,in4);
  int n=analogRead(A0)/4;
  _mSetSpeed(speedPinA,n);
  _mSetSpeed(speedPinB,n);
}
void _mRight(int pin1,int pin2) {
  digitalWrite(pin1,HIGH);
  digitalWrite(pin2,LOW);
}
void _mLeft(int pin1,int pin2) {
  digitalWrite(pin1,LOW);
  digitalWrite(pin2,HIGH);
}
void _mStop(int pin1,int pin2) {
  digitalWrite(pin1,HIGH);
  digitalWrite(pin2,HIGH);
}
void _mSetSpeed(int pinPWM,int SpeedValue) {
  analogWrite(pinPWM,SpeedValue);
}

```

Vorgang

1. Bereiten Sie eine Skizze mit dem gegebenen Code vor und laden Sie sie auf das Board hoch.
2. Schließen Sie das Modul/Komponente anhand des Anschlussdiagramms oder der Anschlusstabelle an.
3. Wenn Sie Ihre Plattform mit Strom versorgen, werden die Pins entsprechend dem Code gesteuert.

Anwendungsbeispiel - 2

Das Beispiel zeigt die Verwendung eines 2-Phasen-Schrittmotors.

Die Anweisungen basieren auf der Arduino®-Plattform. Sie können auch ein Arduino Derivat oder eine andere Plattform verwenden, die das Produkt unterstützt.

Anschluss

Modul	IN1	IN2	IN3	IN4
Arduino®	8	9	10	11

Code

```
#include <Stepper.h>
#define STEPS 100

Stepper stepper(STEPS, 8, 9, 10, 11);
int previous = 0;
void setup() {
  stepper.setSpeed(90);
}
void loop() {
  int val = analogRead(0);
  stepper.step(val - previous);
  previous = val;
}
```

Vorgang

1. Bereiten Sie eine Skizze mit dem gegebenen Code vor und laden Sie sie auf das Board hoch.
2. Schließen Sie das Modul/Komponente anhand des Anschlussdiagramms oder der Anschlussstabelle an.
3. Wenn Sie Ihre Plattform mit Strom versorgen, dreht sich ein angeschlossener Schrittmotor wie im Code festgelegt.

Technische Daten

Treiber	L298N
Betriebsspannung	+5 V/DC bis +35 V/DC
Treiber Eingangsstrom	2 A
Spannungsausgang Logikkreis (Vss)	+5 V/DC bis 7 V/DC (interne Speisung +5 V/DC)
Stromaufnahme Logikkreis	0 - 36 mA
Steuer-Level	Niedrig: -0,3 V/DC bis 1,5 V/DC Hoch: 2,3 V/DC bis Vss
Freischaltung Steuersignal	Niedrig: -0,3 V/DC bis 1,5 V/DC Hoch: 2,3 V/DC -Vss
Leistung (max.)	25 W (bei 75 °C)
Betriebstemperatur	-25 bis +130°C
Abmessungen (ca.)	60 x 46 x 30 mm
Gewicht (ca.)	30 g

Entsorgung



Elektronische Geräte sind Wertstoffe und gehören nicht in den Hausmüll.

Entsorgen Sie das Produkt am Ende seiner Lebensdauer gemäß den geltenden gesetzlichen Bestimmungen.

Sie erfüllen damit die gesetzlichen Verpflichtungen und leisten Ihren Beitrag zum Umweltschutz.

Impressum

Dies ist eine Publikation der Conrad Electronic SE, Klaus-Conrad-Str. 1, D-92240 Hirschau (www.conrad.com).

Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktionen jeder Art, z. B. Fotokopie, Mikroverfilmung, oder die Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers. Nachdruck, auch auszugsweise, verboten.

Die Publikation entspricht dem technischen Stand bei Drucklegung.
Copyright 2020 by Conrad Electronic SE.



BN 2134135

Stepper Motor Driver Module

EN Quick Instructions

Version: 2

Delivery Content

- Product

Description

The double H driver is a high voltage/high current dual full-bridge driver with a L298N at its core, an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages.

It is designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors.

Each driver has an enable input, which allows you to enable/disable the driver independently of the input signals.

The emitters of the lower transistors of each bridge are connected together.

Use the matching external terminals to connect an external sensing resistor.

The additional voltage supply ensures the logic can work at a lower voltage.

Development Resources

Development resources and additional product information are available from:



<https://www.conrad.de/>

Pinout / Pin Map

Pin Name	Description
MOTORA	Motor1
MOTORB	Motor2
VMS	5 V/DC - 35 V/DC
GND	Ground
5V	Power input for the logic circuit on the board
ENA	Motor 1 enable pin

Pin Name	Description
IN1	Motor 1 control pin
IN2	Motor 1 control pin
ENB	Motor 2 enable pin
IN3	Motor 2 control pin
IN4	Motor 2 control pin
5V	Output 5V
GND	Ground
CSB	Current test pin for motor 2: Wire to a resistor for current testing or disable through a jumper.
CSA	Current test pin for motor 1: Wire to a resistor for current testing or disable through a jumper.
UR1	Pull up resistor
UR2	Pull up resistor
UR3	Pull up resistor
UR4	Pull up resistor
5V_EN	5V source jumper: Jumper on: 5V supply to logic circuit via VMS port ($7V < VMS < 18V$) Jumper removed: 5V supply to logic circuit via 5V port

Example Application - 1

This example demonstrates the use of a 2 Channel DC Motor.

The instructions use the Arduino® platform to illustrate product use. You can also use an Arduino derivative or another platform that supports this type of product.

Connection

Module	IN1	IN2	IN3	IN4
Arduino®	13	12	11	10

Code

```
int in1=13;
int in2=12;
int in3=11;
int in4=10;
int speedPinA=6;
int speedPinB=5;
```

```

void setup() {
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  digitalWrite(in1,HIGH);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,HIGH);
}
void loop() {
  _mRight(in1,in2);
  _mRight(in3,in4);
  int n=analogRead(A0)/4;
  _mSetSpeed(speedPinA,n);
  _mSetSpeed(speedPinB,n);
}
void _mRight(int pin1,int pin2) {
  digitalWrite(pin1,HIGH);
  digitalWrite(pin2,LOW);
}
void _mLeft(int pin1,int pin2) {
  digitalWrite(pin1,LOW);
  digitalWrite(pin2,HIGH);
}
void _mStop(int pin1,int pin2) {
  digitalWrite(pin1,HIGH);
  digitalWrite(pin2,HIGH);
}
void _mSetSpeed(int pinPWM,int SpeedValue) {
  analogWrite(pinPWM,SpeedValue);
}

```

Procedure

1. Prepare a sketch with the given code and upload it to your board.
2. Connect the module/component to the board as shown in the connection diagram or table.
3. When you supply your platform with power the pins are controlled as determined by the code.

Example Application - 2

The example demonstrates the use of a 2 phase stepper motor.

The instructions use the Arduino® platform to illustrate product use. You can also use an Arduino derivative or another platform that supports this type of product.

Connection

Module	IN1	IN2	IN3	IN4
Arduino®	8	9	10	11

Code

```
#include <Stepper.h>
#define STEPS 100

Stepper stepper(STEPS, 8, 9, 10, 11);
int previous = 0;
void setup() {
  stepper.setSpeed(90);
}
void loop() {
  int val = analogRead(0);
  stepper.step(val - previous);
  previous = val;
}
```

Procedure

1. Prepare a sketch with the given code and upload it to your board.
2. Connect the module/component to the board as shown in the connection diagram or table.
3. When you supply your platform with power a connected stepper motor steps as determined in the code.

Specifications

Driver	L298N
Operating voltage	+5 V/DC to +35 V/DC
Driver I _o	2 A
Logic power output V _{ss}	+5 V/DC to 7 V/DC (internal supply +5 V/DC)
Logic current	0 - 36 mA
Controlling level	Low -0.3 V/DC to 1.5 V/DC High: 2.3 V/DC to V _{ss}
Enable signal level	Low -0.3 V/DC to 1.5 V/DC High: 2.3 V/DC -V _{ss}
Power (max.)	25 W (at 75 °C)
Working temperature	-25 to +130°C
Dimensions (approx.)	60 x 46 x 30 mm
Weight (approx.)	30 g

Disposal



Electronic devices are recyclable waste and must not be disposed of in the household waste.

At the end of its service life, dispose of the product in accordance with applicable regulatory guidelines.

You thus fulfill your statutory obligations and contribute to the protection of the environment.

Legal Notice

This is a publication by Conrad Electronic SE, Klaus-Conrad-Str. 1, D-92240 Hirschau (www.conrad.com).

All rights including translation reserved. Reproduction by any method, e.g. photocopy, microfilming, or the capture in electronic data processing systems require the prior written approval by the editor.

Reprinting, also in part, is prohibited.

This publication represent the technical status at the time of printing.

Copyright 2020 by Conrad Electronic SE.

