

joy-it



Ramps 1.4

Index

1. Einführung
2. Anschluss
 - 2.1 Schematische Darstellung
 - 2.2 Verbindung zum Arduino MEGA
 - 2.3 Jumper-Konfiguration
 - 2.4 Schrittmotor-Treiber
 - 2.5 EndStops und Temperatursensoren
 - 2.6 Stromversorgung und Display
3. Einrichtung der Entwicklungsumgebung
4. Konfiguration der Software
 - 4.1 Boardauswahl
 - 4.2 Display
 - 4.3 Extruderanzahl
 - 4.4 Temperatursensoren
 - 4.5 Drehrichtungen und Schrittmotoren
5. Support

1. Einführung

Sehr geehrter Kunde,

vielen Dank, dass sie sich für unser Produkt entschieden haben.

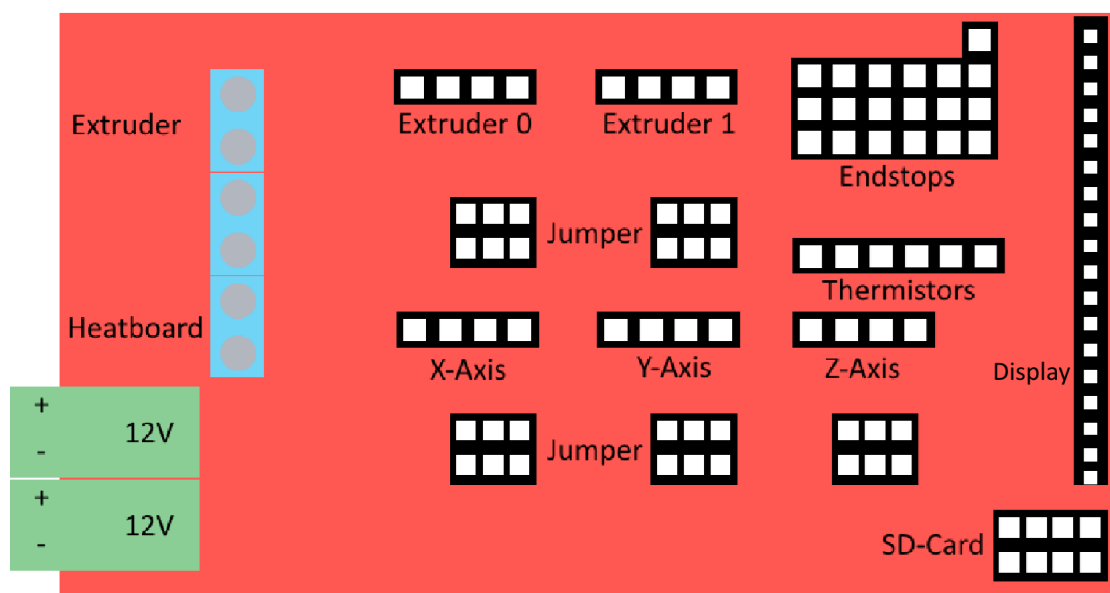
Im folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

2. Anschluss

2.1 Schematische Darstellung

Der folgenden schematischen Darstellung können Sie eine detaillierte Anschlussbelegung entnehmen.

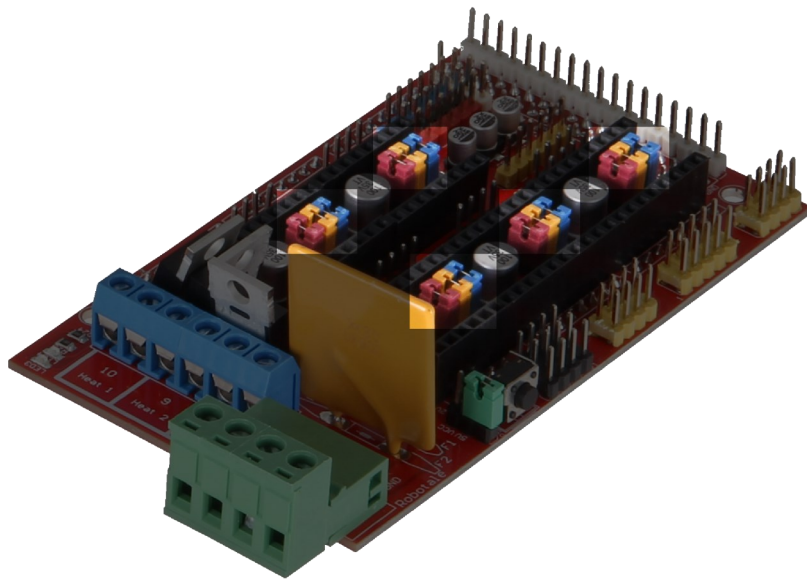


2.2 Verbindung zum Arduino MEGA

Beginnen Sie zuerst damit, das Ramps 1.4 Erweiterungsboard an Ihren Arduino MEGA anzuschließen. Stecken Sie das Board einfach auf Ihren Arduino auf, sodass die Pins auf der Unterseite des Boards in die entsprechenden Anschlüsse des Arduinos geführt werden.

2.3 Jumper-Konfiguration

Bevor die nötigen Schrittmotoren auf dem Erweiterungsboard aufgesteckt werden können, ist es nötig die jeweiligen Jumper zu setzen. Die Jumper definieren die Anzahl der möglichen Microschritte für die Schrittmotoren. Es werden bis zu 1/128 Schritte unterstützt. Die genaue Konfiguration ist zusätzlich abhängig von den verwendeten Schrittmotortreibern.

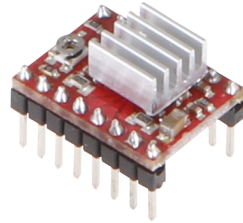


Die verschiedenen Jumper-Steckplätze sind auf dem oberen Bild, sowie auf der schematischen Darstellung auf der vorherigen Seite zu sehen.

Die unterschiedlichen Schrittkonfigurationen sind der folgenden Tabelle zu entnehmen:

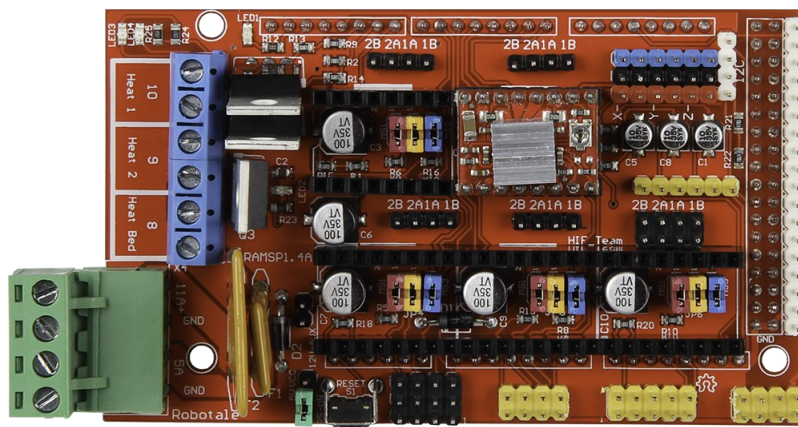
| Jumper 1 | Jumper 2 | Jumper 3 | Schritte |
|----------|----------|----------|----------------|
| Nein | Nein | Nein | Ganzer Schritt |
| Ja | Nein | Nein | Halber Schritt |
| Nein | Ja | Nein | 1/4 Schritt |
| Ja | Ja | Nein | 1/8 Schritt |
| Nein | Nein | Ja | 1/16 Schritt |
| Ja | Nein | Ja | 1/32 Schritt |
| Ja | Ja | Ja | 1/128 Schritt |

2.4 Schrittmotor-Treiber



Nachdem die Jumper gesetzt wurden, können die Schrittmotortreiber in die schwarzen Steckplätze, welche sich direkt an den Jumpersteckplätzen befinden, eingesetzt werden. Achten Sie bitte darauf die Schrittmotortreiber richtig einzusetzen. Die Beschriftungen an den Pins der Treiber sollten mit den Beschriftungen auf dem Ramps-Erweiterungsboard übereinstimmen.

Nach dem erfolgreichen Einsatz der Motortreiber können nun die entsprechenden Schrittmotoren angeschlossen werden. Setzen Sie die A9488 Motortreiber wie im folgendem Bild ein.



2.5 EndStops und Temperatursensoren

Nun können die Endstops und die Temperatursensoren angeschlossen werden.

Hier ist ein Anschluss von bis zu 6 Endstops und 3 Temperatursensoren angeschlossen werden.

Die genaue Anschluss der Endstops kann auch [hier](#) eingesehen werden.

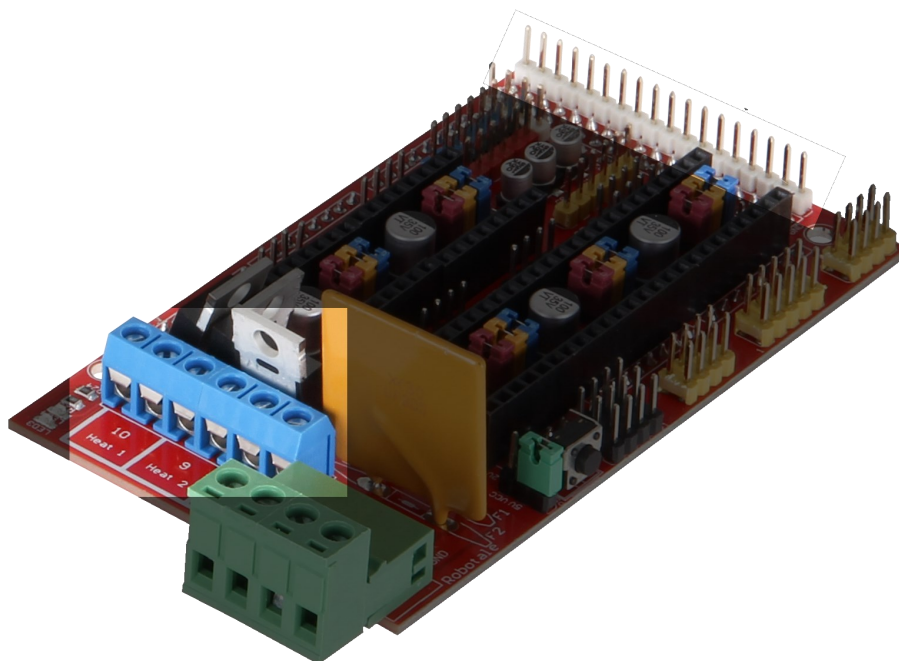
Bei den Temperatursensoren wird der Anschluss **T0** für das HotEnd und der Anschluss **T1** für das Heizbett verwendet.

Die Polung der Temperatursensoren ist nicht relevant.

2.6 Stromversorgung und Display

Zusätzlich wird nun an den Anschluss **D8** das Heizbett und an den Anschluss **D10** das HotEnd angeschlossen.

D9 kann entweder für einen Bauteilkühhlfüfter oder für ein zusätzliches HotEnd verwendet werden. Auch das Display kann nun an dem dafür vorgesehenen Steckplatz angeschlossen werden. Abschließend muss nun das Erweiterungsboard noch mit Strom versorgt werden.



3. Einrichtung der Entwicklungsumgebung

Für die Verwendung benötigen Sie die Arduino-Entwicklungsumgebung, sowie die Marlin 3D Printer-Firmware.

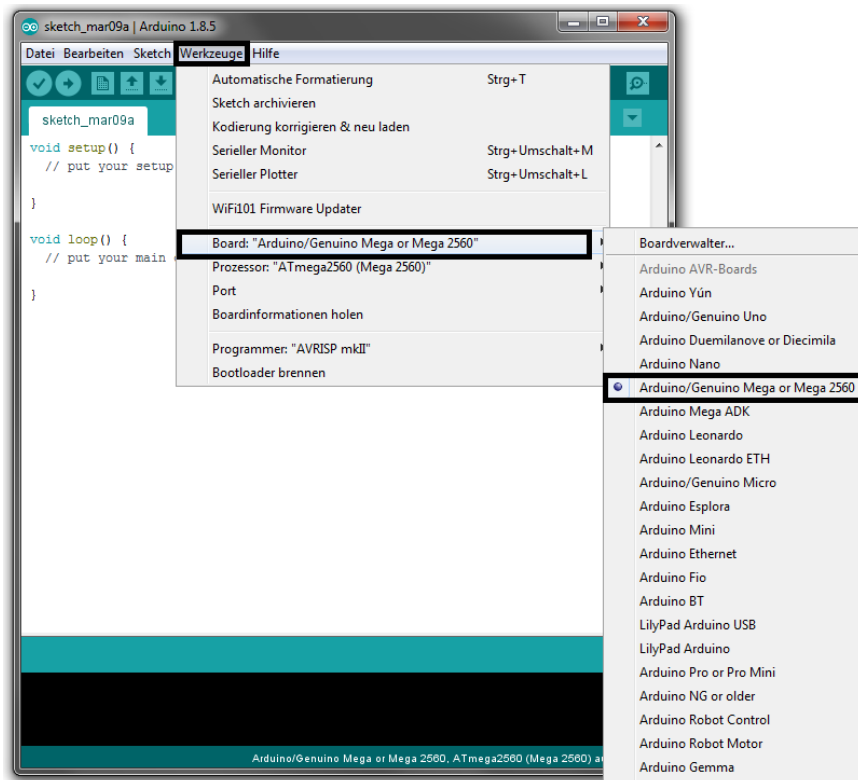
Laden Sie sich zunächst die aktuelle Arduino-Software [hier](#) herunter und installieren Sie diese.

Anschließend können Sie sich die aktuelle Marlin Firmware [hier](#) herunterladen.

Öffnen Sie mit der Arduino Entwicklungsumgebung einfach die „Marlin.ino“-Datei aus dem heruntergeladenen Marlin-Firmware-Verzeichnis.

Die Entwicklungsumgebung wird daraufhin die Firmware laden und alle notwendigen Dateien öffnen.

Wechseln Sie nun auf den „Arduino Mega 2560“ als ausgewählter Board-Typ, damit die Übertragung der Firmware auf das Board ermöglicht wird:



4. Konfiguration der Software

4.1 Boardauswahl

Bevor die Software auf den Arduino übertragen werden kann, muss diese zunächst auf die verwendete Konfiguration angepasst werden.

Für das Ramps 1.4 Board sind in der Marlin Software die folgenden Konfigurationen vorgesehen:

```
#define BOARD_RAMPS_14_EFB 43 // RAMPS 1.4 (Power outputs: Hotend, Fan, Bed)
#define BOARD_RAMPS_14_EEB 44 // RAMPS 1.4 (Power outputs: Hotend0, Hotend1, Bed)
#define BOARD_RAMPS_14_EFF 45 // RAMPS 1.4 (Power outputs: Hotend, Fan0, Fan1)
#define BOARD_RAMPS_14_EEF 46 // RAMPS 1.4 (Power outputs: Hotend0, Hotend1, Fan)
#define BOARD_RAMPS_14_SF 48 // RAMPS 1.4 (Power outputs: Spindle, Controller Fan)
```

Die häufigste Konfiguration besteht aus einem HotEnd, einem Fan und einem Bed.

Je nach Einsatzzweck und Aufbau kann diese Konfiguration bei Ihnen allerdings abweichen.

Die entsprechende Konfiguration muss nun als Motherboard in der **Configurations.h** Datei eingetragen werden.

Öffnen Sie dazu die **Configurations.h** Datei und suchen Sie hier nach der entsprechenden Motherboard-Definition.

```
// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_RAMPS_14_EFB
#endif
```

Für die oben erwähnte Konfiguration (HotEnd, Fan, Bed) sollte die Motherboard-Konfiguration wie folgt aussehen:

4.2 Display

Für die Verwendung des Displays muss eine zusätzliche Konfiguration vorgenommen werden.

```
/**
 * LCD TYPE
 *
 * Enable ULTRA_LCD for a 16x2, 16x4, 20x2, or 20x4 character-based LCD.
 * Enable DOGLCD for a 128x64 (ST7565R) Full Graphical Display.
 * (These options will be enabled automatically for most displays.)
 *
 * IMPORTANT: The U8glib Library is required for Full Graphic Display!
 *             https://github.com/olikraus/U8glib_Arduino
 */
#define ULTRA_LCD    // Character based
#define DOGLCD        // Full graphics display
```

Suchen Sie dafür, ebenfalls in der **configuration.h** Datei nach der folgenden Sektion:

```
#define DOGLCD        // Full graphics display
```

Um das Display zu aktivieren, entfernen Sie aus der letzten Zeile die Kommentarstriche, sodass die Zeile wie folgt aussieht:

Außerdem ist für die Verwendung des Displays zusätzlich die U8glib-Bibliothek erforderlich. Laden Sie dazu die Bibliothek [hier](#) herunter und kopieren Sie diese in Ihren Arduino-Bibliotheksordner (C:\Users\[Ihr Benutzername]\Documents\Arduino\libraries).

```
// This defines the number of extruders
#define EXTRUDERS 1
```

Sollten Sie mehr als einen Extruder verwenden, so muss außerdem die folgende Zeile dementsprechend angepasst werden:

4.3 Extruderanzahl

Sollten Sie mehr als einen Extruder verwenden, so muss außerdem die folgende Zeile in der **configuration.h** Datei dementsprechend angepasst werden:

```
// This defines the number of extruders  
#define EXTRUDERS 1
```

4.4 Temperatursensoren

Auch eine Konfiguration der Temperatursensoren kann, je nach verwendeten Sensoren, notwendig sein. Beachten Sie dafür die Konfigurationshinweise in folgender Sektion der **configurations.h** Datei:

```
#define TEMP_SENSOR_0 1  
#define TEMP_SENSOR_1 0  
#define TEMP_SENSOR_2 0  
#define TEMP_SENSOR_3 0  
#define TEMP_SENSOR_4 0  
#define TEMP_SENSOR_BED 0
```

Auch die maximal zulässigen Temperaturen können, je nach verwendeten Bauteilen, abweichen. Stimmen Sie daher die maximalen Temperaturen in dem folgenden Bereich auf ihre Bauteile ab:

```
// When temperature exceeds max temp, your heater will be switched off.  
// This feature exists to protect your hotend from overheating accidentally,  
but *NOT* from thermistor short/failure!  
// You should use MINTEMP for thermistor short/failure protection.  
#define HEATER_0_MAXTEMP 275  
#define HEATER_1_MAXTEMP 275  
#define HEATER_2_MAXTEMP 275  
#define HEATER_3_MAXTEMP 275  
#define HEATER_4_MAXTEMP 275  
#define BED_MAXTEMP 150
```

4.5 Drehrichtungen der Schrittmotoren

Je nach Verkabelung bzw. Polung der verwendeten Schrittmotoren, kann es notwendig sein die Drehrichtung einzelner Motoren zu invertieren.

Dies ist in der folgenden Sektion möglich:

```
// Invert the stepper direction. Change (or reverse the motor connector) if an  
axis goes the wrong way.  
#define INVERT_X_DIR false  
#define INVERT_Y_DIR true  
#define INVERT_Z_DIR false  
  
// Enable this option for Toshiba stepper drivers  
// #define CONFIG_STEPPERS_TOSHIBA  
  
// @section extruder  
  
// For direct drive extruder v9 set to true, for geared extruder set to false.  
#define INVERT_E0_DIR false  
#define INVERT_E1_DIR false  
#define INVERT_E2_DIR false  
#define INVERT_E3_DIR false  
#define INVERT_E4_DIR false
```

5. Support

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

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

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

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net