



# ROBOTER Bausatz-Shop

## MKS GenL Datenblatt

## Inhaltsverzeichnis

Technische Daten .....	3
Features .....	3
Beschreibung.....	3
Schaltplan und Anschlüsse .....	4
Schaltplan .....	4
Anschlussplan .....	5
Verkabelung.....	6
Abmessungen .....	7
Schrittmotortreibereinstellungen .....	8
Anschluss der Schrittmotorentreiber .....	8
Einstellen der Mikroschritte .....	8
Firmware Setup .....	9
Firmware .....	9
Marlin 2.0 Firmware kompilieren und uploaden mit Platform-io.....	9

## Technische Daten

- Modell: MKS Gen L
- CPU: ATMEGA2560-AU, 8bit 16MHZ, 5V Logikpegel
- Serialport Chip: CH340G
- Betriebsspannung: 12-24V

## [Treiber-Download für den verwendeten CH340-Chip](#)

## Features

- Solides Layout mit optimierter Störfestigkeit
- Firmware kann die selbe Konfiguration benutzen wie RAMPS 1.4
- 3 Anschlüsse für Thermistoren
- bis zu 5 Schrittmotoren mit Pololu-Treibern z. B. A4988, DRV8825, TMC2100, LV8729, uvm.
- Unterstützt externe Signale für 57 und 87er Schrittmotoren
- 4x PWM-fähige Mosfet Outputs (Heizbett, Extruder0, Extruder1, Fans)
- 4 Lagen PCB zur besseren Wärmeverteilung
- 6 Endschalter-Anschlüsse
- Sicherung zur Verhinderung von Kurzschlüssen
- Farbige Anschlüsse zum einfachen Verkabeln

## Beschreibung

Das MKS Gen L Board ist eine kompakte All-in-One Steuereinheit für RepRap 3D-Drucker und andere CNC Geräte.

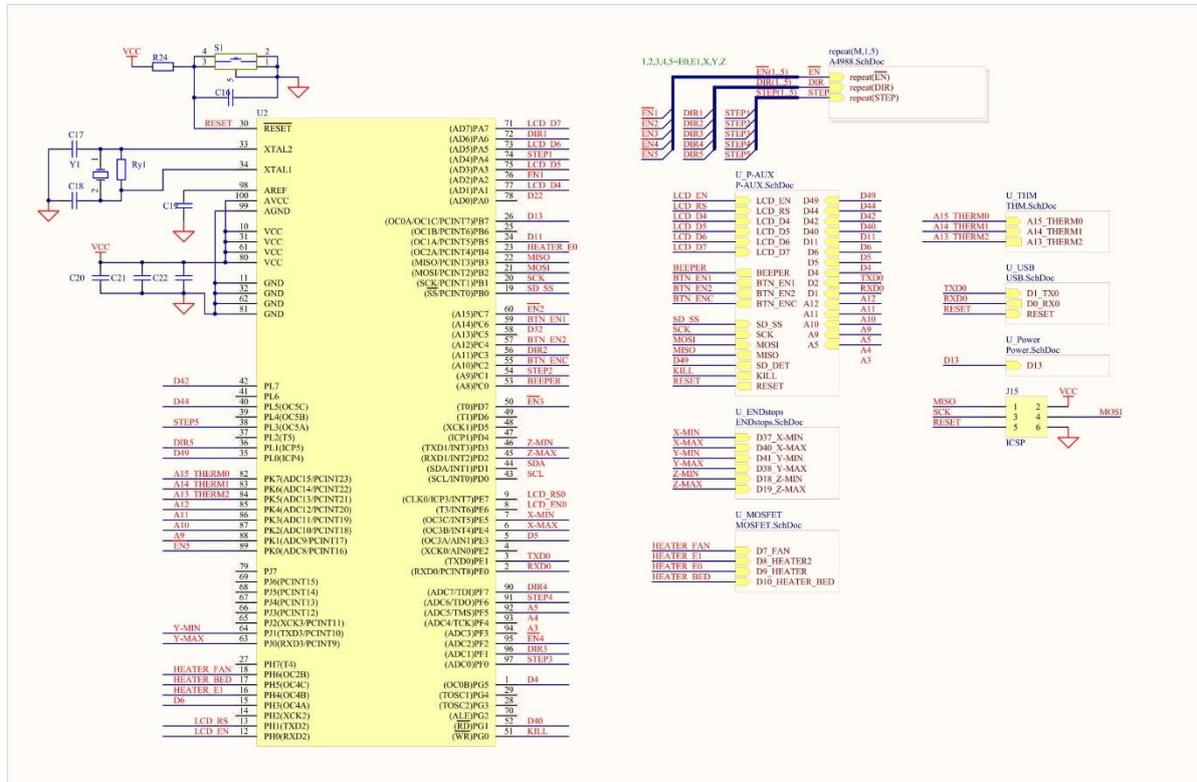
Das MKS Gen L ist mit dem Ziel entwickelt worden dem Benutzer eine flexible, entwicklungsfreundliche Elektronik zur Steuerung von 3D-Druckern zu bieten. Es besitzt einen onboard ATmega2560 und bietet dieselben Programmiermöglichkeiten wie das Ramps 1.4 Board und kann sogar mit derselben Konfiguration arbeiten.

Das MKS Gen L Board besitzt dem MKS Base gegenüber den Vorteil, dass Pololu-kompatible Schrittmotorentreiber verwendet werden können. Je nach Präferenz lassen sich so Setups mit A4988, DRV8825, TMC2100 oder andere Schrittmotorentreibern installieren.

Das Board kann mit einer Spannung von 12V-24V betrieben werden und eignet sich auch für Dual Extruder.

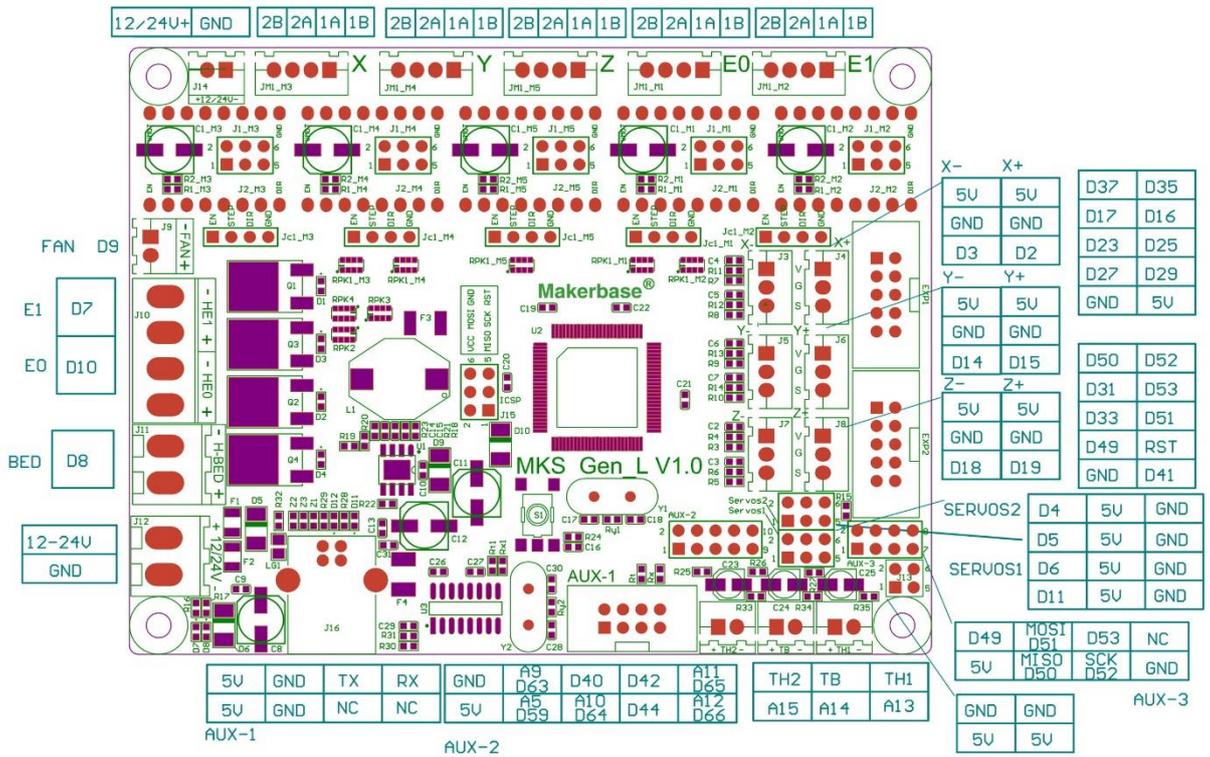
# Schaltplan und Anschlüsse

## Schaltplan

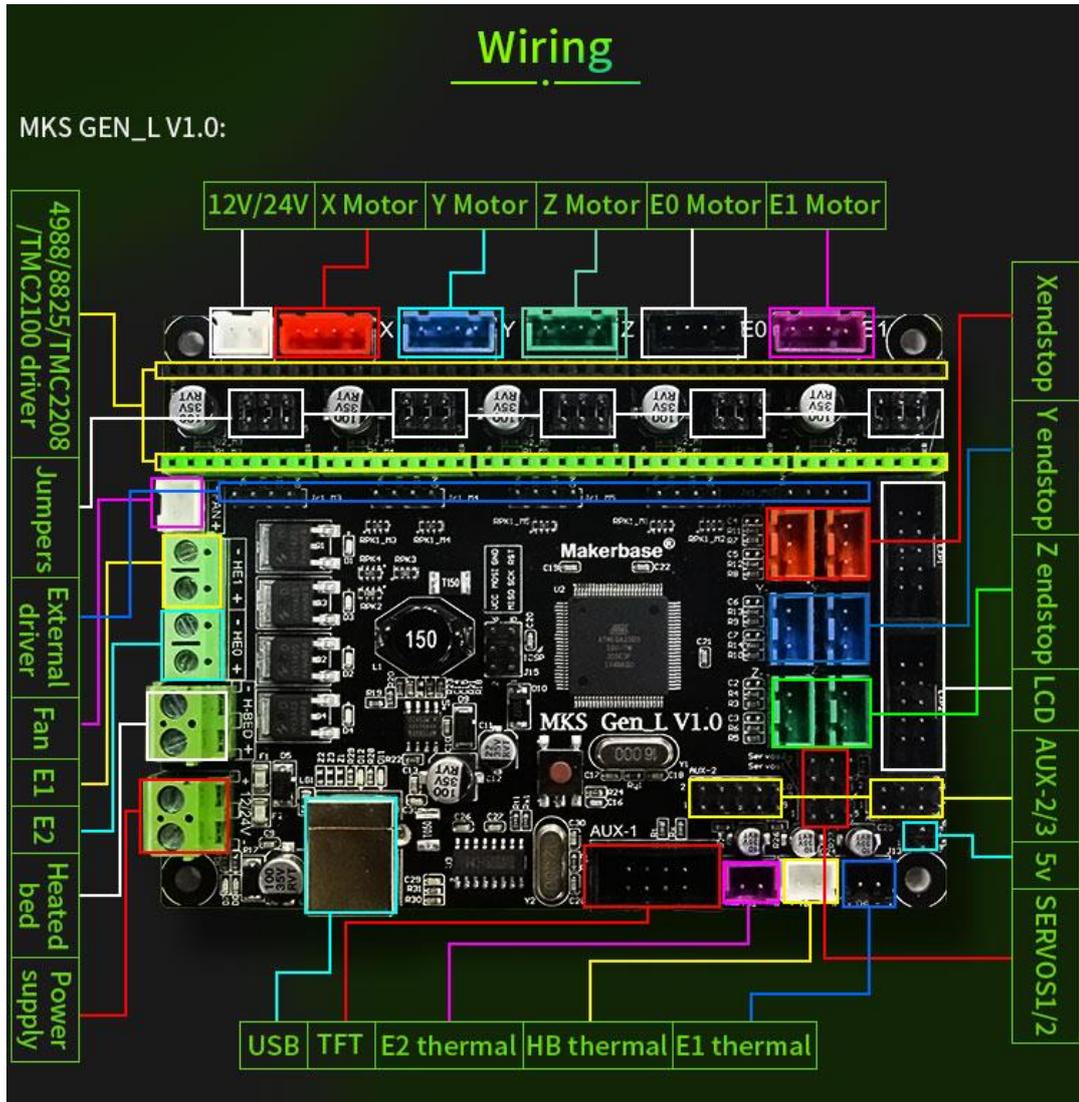


Den kompletten Schaltplan finden Sie [hier zum Download](#).

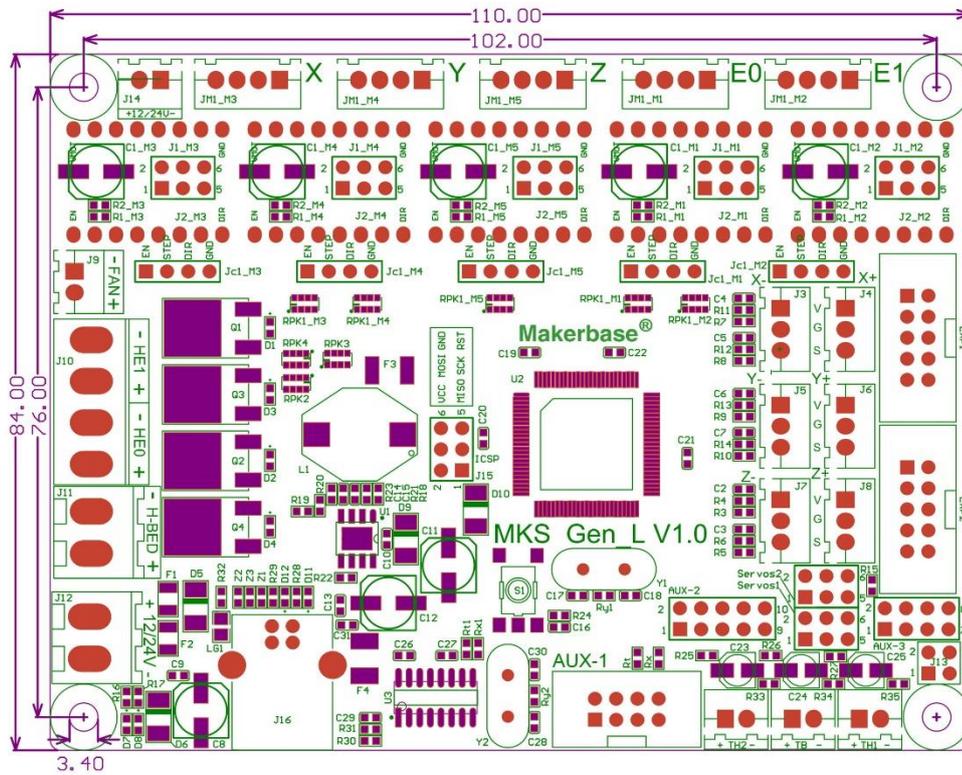
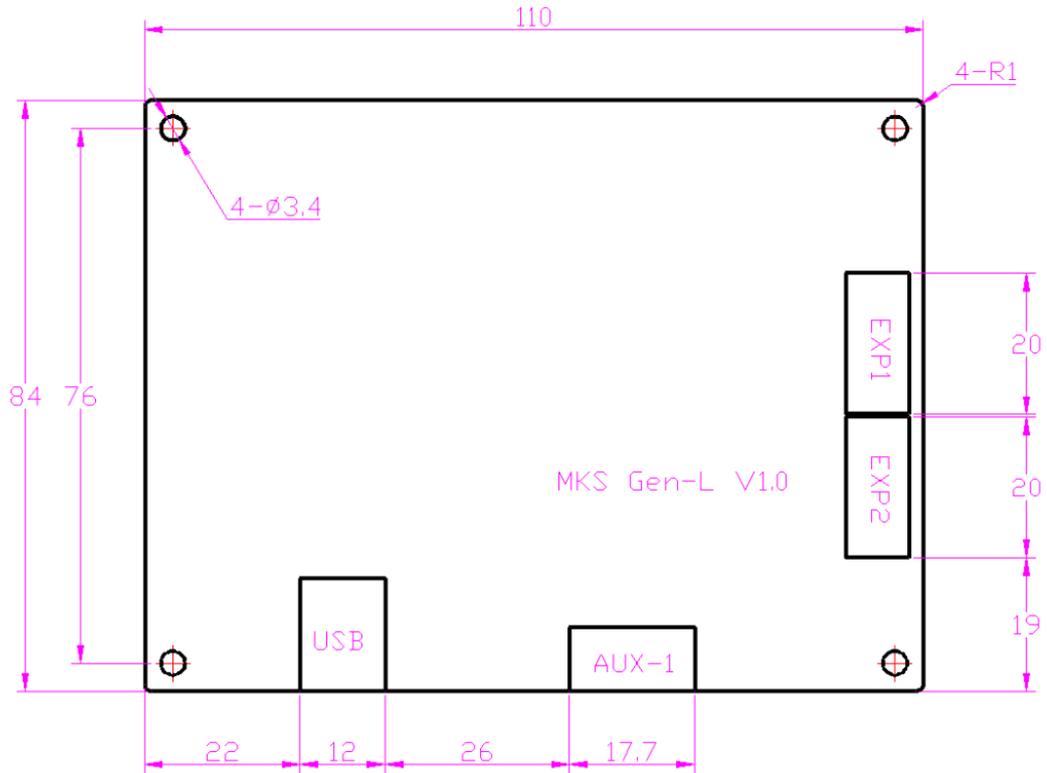
# Anschlussplan



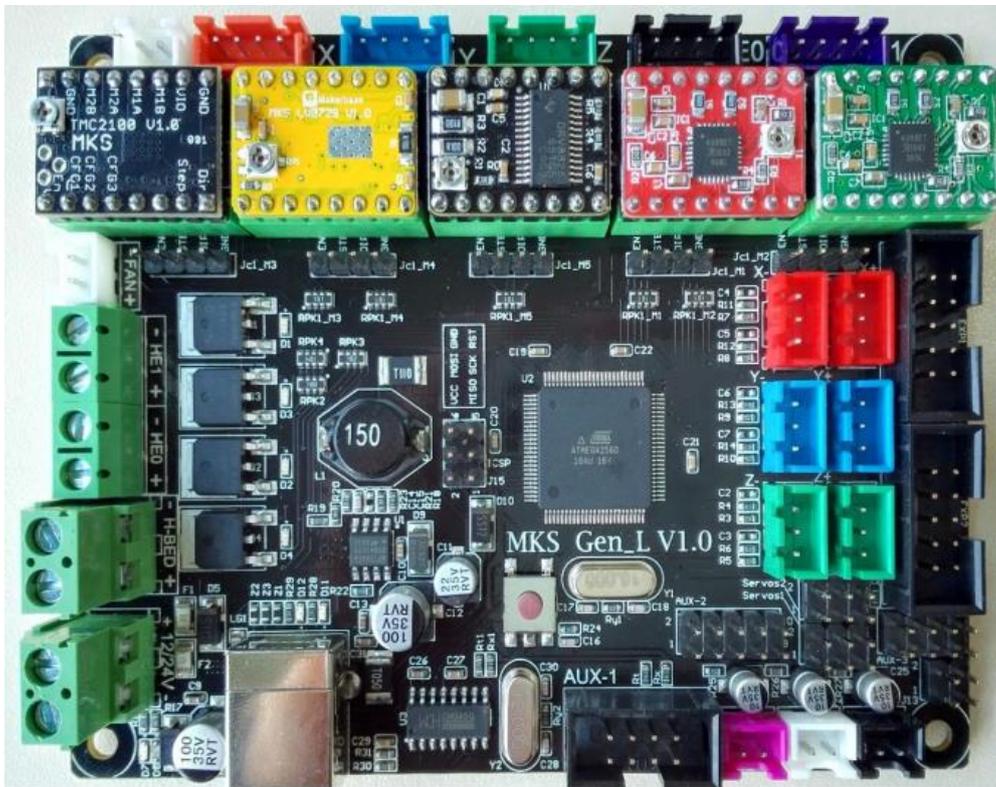
Verkabelung



## Abmessungen



## Schrittmotortreibereinstellungen Anschluss der Schrittmotorentreiber



### Einstellen der Mikroschritte

A4988				DRV8825				LV8729			
M1	M2	M3	Step	M1	M2	M3	Step	M1	M2	M3	Step
Low	Low	Low	Full	Low	Low	Low	Full	Low	Low	Low	Full
High	Low	Low	1/2	High	Low	Low	1/2	High	Low	Low	1/2
Low	High	Low	1/4	Low	High	Low	1/4	Low	High	Low	1/4
High	High	Low	1/8	High	High	Low	1/8	High	High	Low	1/8
High	High	High	1/16	Low	Low	High	1/16	Low	Low	High	1/16
				High	Low	High	1/32	High	Low	High	1/32
				Low	High	High	1/32	Low	High	High	1/64
				High	High	High	1/32	High	High	High	1/128

Die Jumper sind von links nach rechts M1, M2, M3. Eingesteckt ist „High“, wird der Jumper entfernt ist das Signal „Low“.

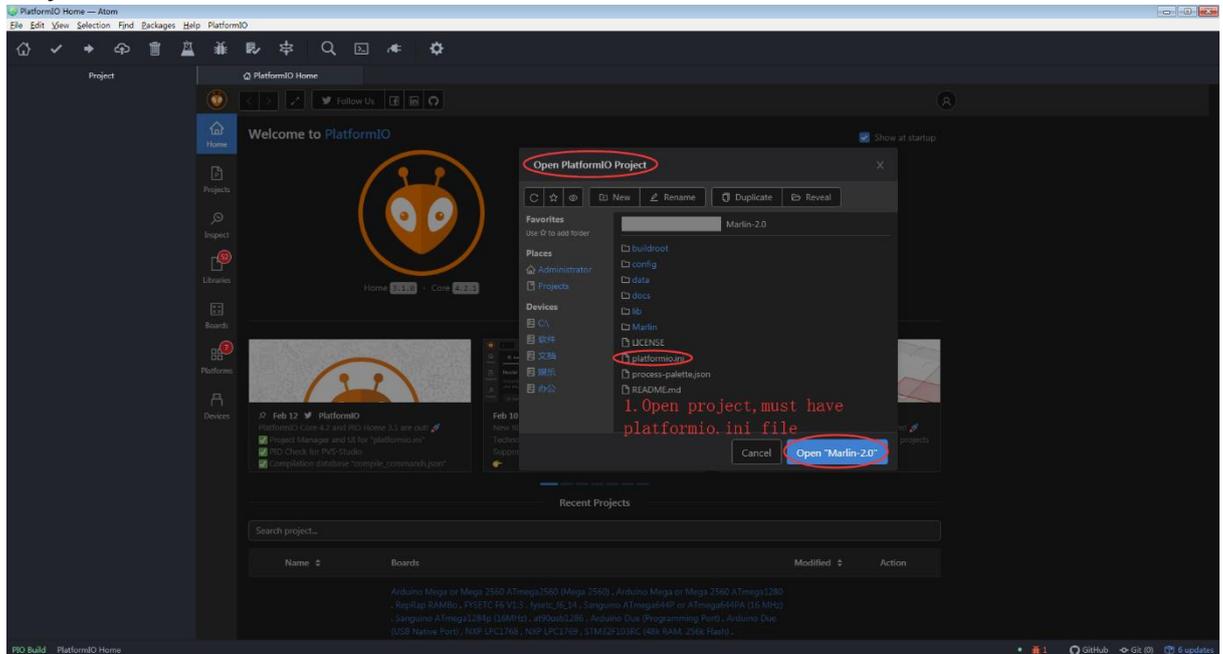
## Firmware Setup

### Firmware

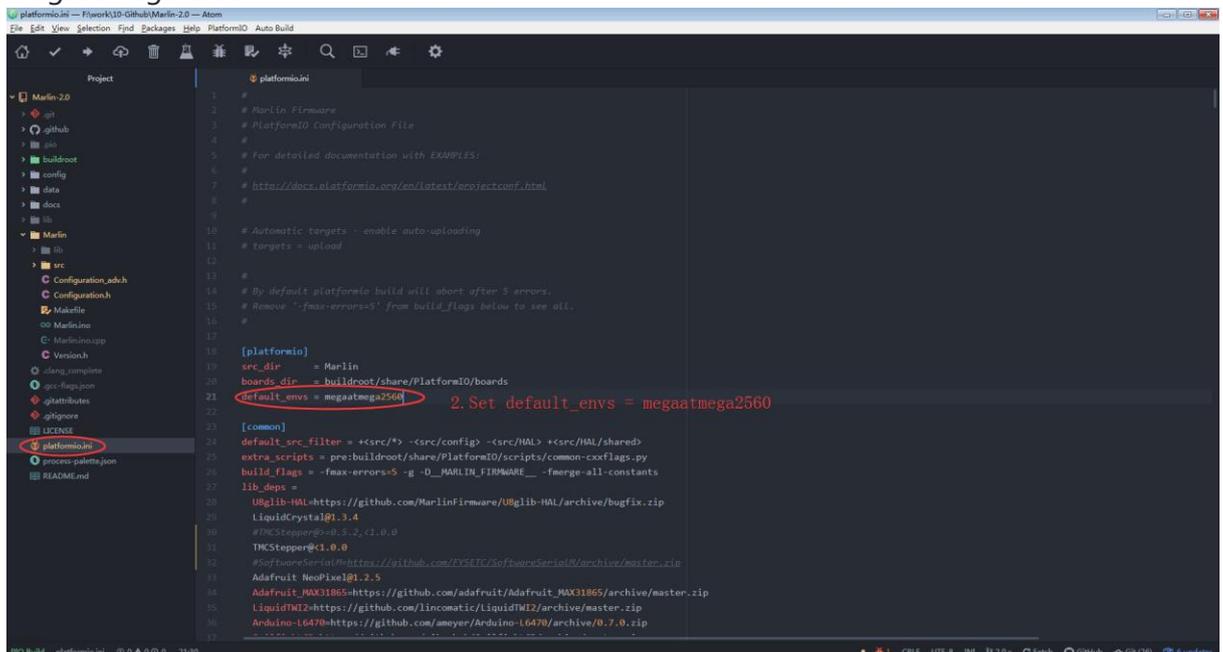
#### [Marlin 2.0 Firmware Download](#)

### Marlin 2.0 Firmware kompilieren und uploaden mit Platform-io

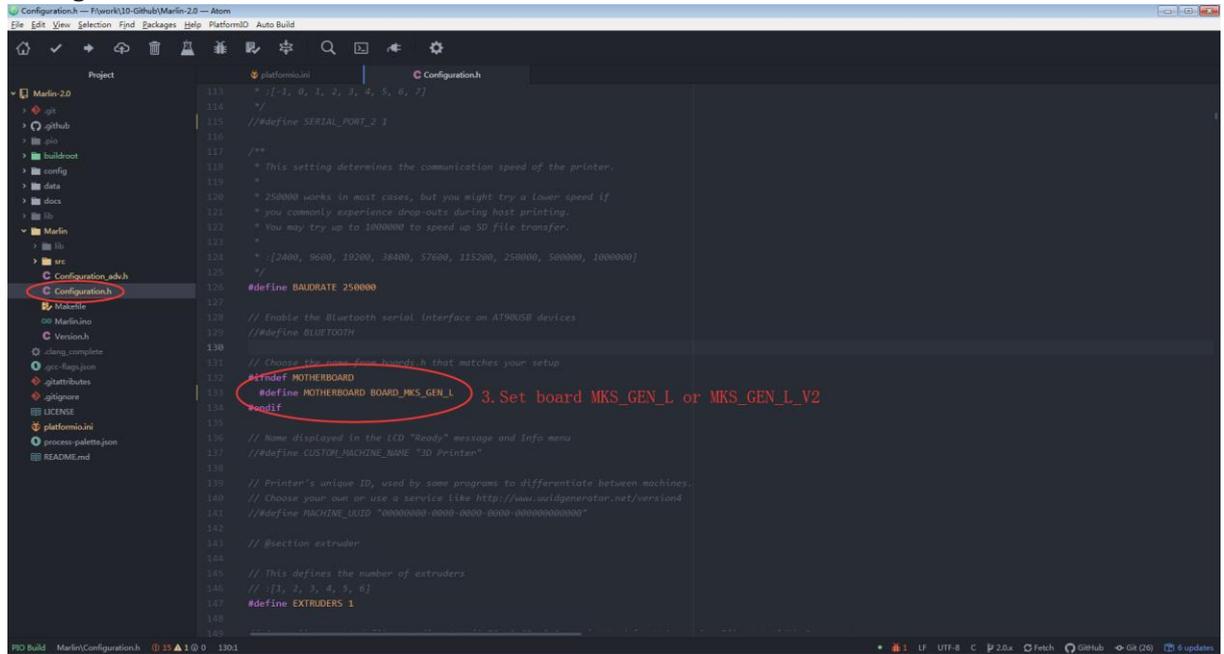
#### 1. Projekt öffnen



#### 2. Platformio.ini Datei öffnen und ATmega2560 einstellen: set default\_envs = megaatmega2560



### 3. configuration.h Datei öffnen, Board als MKS\_GEN\_L oder MKS\_GEN\_L\_V2 einstellen

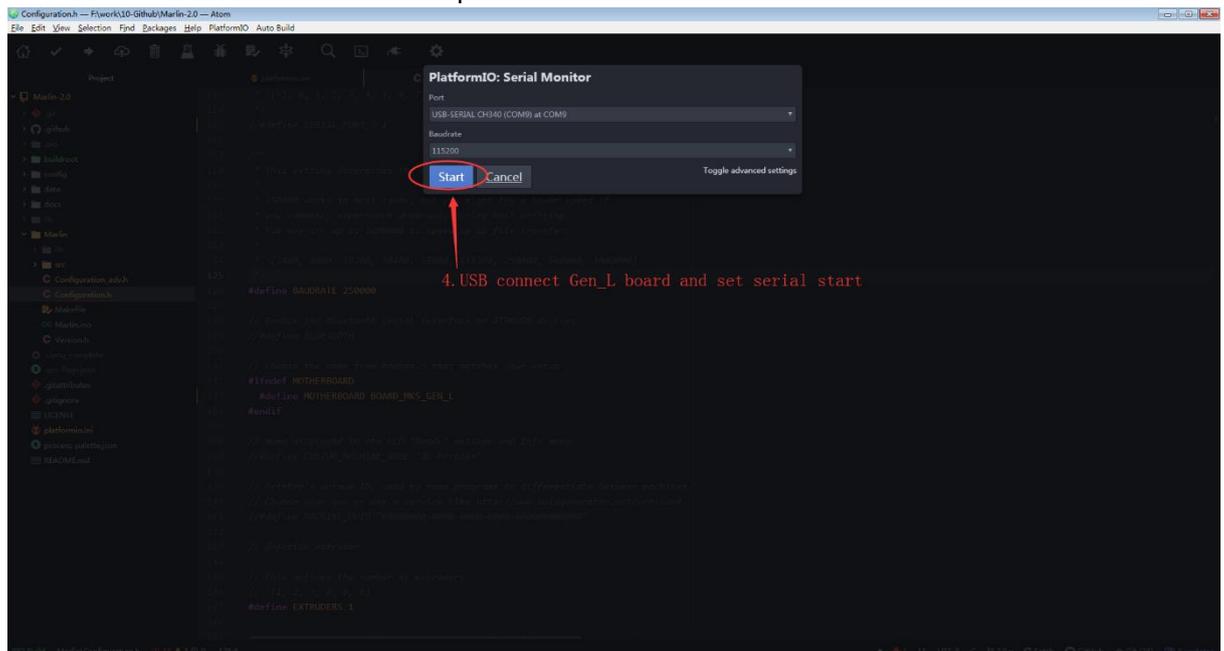


```

113 * [-1, 0, 1, 2, 3, 4, 5, 6, 7]
114 */
115 // #define SERIAL_PORT_2 1
116
117 /**
118  * This setting determines the communication speed of the printer.
119  *
120  * 250000 works in most cases, but you might try a lower speed if
121  * you commonly experience drop-outs during host printing.
122  * You may try up to 1000000 to speed up SD file transfer.
123  *
124  * [2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
125  */
126 #define BAUDRATE 250000
127
128 // Enable the Bluetooth serial interface on AT90USB devices
129 // #define BLUETOOTH
130
131 // Choose the name from boards.h that matches your setup
132
133 #define MOTHERBOARD
134 #define MOTHERBOARD BOARD_MKS_GEN_L
135 #endif
136
137 // Name displayed in the LCD "Ready" message and Info menu
138 // #define CUSTOM_MACHINE_NAME "3D Printer"
139
140 // Printer's unique ID, used by some programs to differentiate between machines.
141 // Choose your own or use a service like http://www.uuidgenerator.net/version4
142 // #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
143
144 // @section extruder
145 // This defines the number of extruders
146 // [1, 2, 3, 4, 5, 6]
147 #define EXTRUDERS 1
148
149

```

### 4. Board und Baudrate auswählen und per USB verbinden



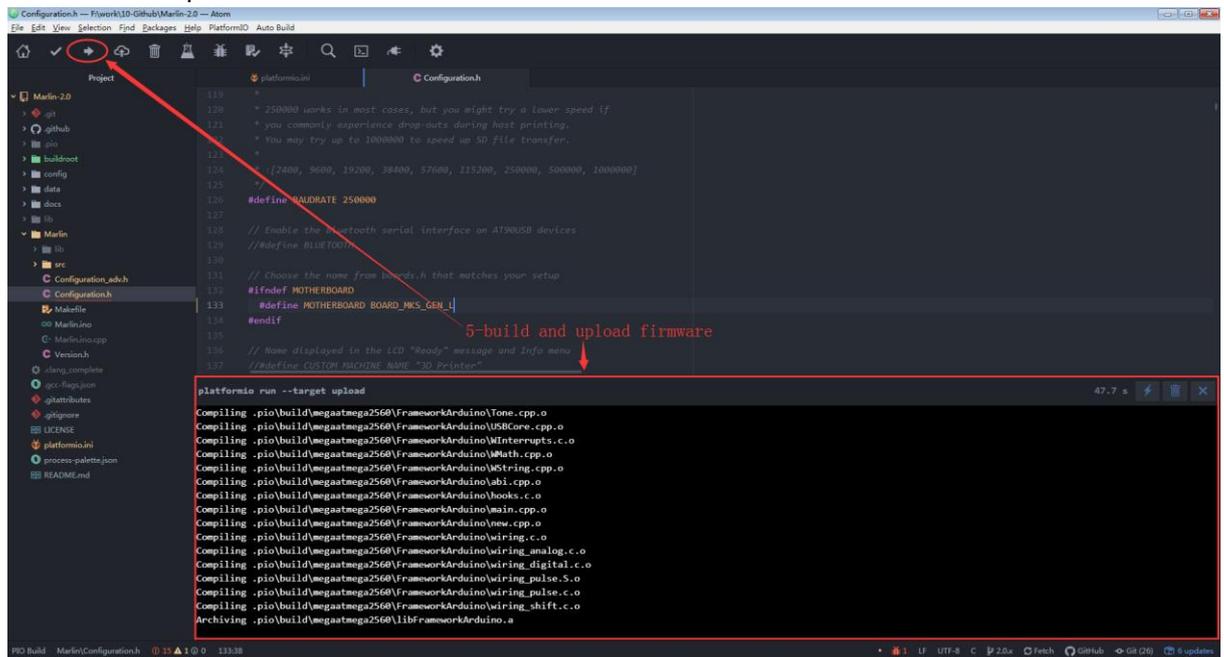
PlatformIO: Serial Monitor

Port: USB-SERIAL CH340 (COM9) at COM9

Baudrate: 115200

Start Cancel Toggle advanced settings

## 5. Firmware kompilieren und hochladen



The screenshot shows the PlatformIO IDE interface. The top toolbar contains a 'Build' button (a play icon) which is highlighted with a red circle and a red arrow. The main editor displays the 'Configuration.h' file for a Marlin 2.0 project, with various preprocessor directives and comments. A red arrow points from the 'Build' button to the terminal window at the bottom, which contains the command 'platformio run --target upload'. The terminal output shows the compilation of various source files, including 'one.cpp', 'usbcore.cpp', 'winterrupts.c', 'math.cpp', 'vstring.cpp', 'abi.cpp', 'hooks.c', 'main.cpp', 'new.cpp', 'wiring.c', 'wiring\_analog.c', 'wiring\_digital.c', 'wiring\_pulse.S', 'wiring\_pulse.c', and 'wiring\_shift.c'. The status bar at the bottom indicates 'PIO Build' and 'Marlin(Configuration.h)'. A red box highlights the terminal output, and a red arrow points to the 'platformio run --target upload' command with the text '5-build and upload firmware'.