

Inhaltsverzeichnis

1. Spider Board V1.0	3
3.3V Problematik	3
Schaubild Verdrahtung	4
Hardwareseitige Konfiguration - UART Verbindung Raspberry Pi	5
Softwareseitige Konfiguration - UART Verbindung Raspberry Pi	6
2. TMC 2209 V3.1 Fysetc	7
Technische Spezifikationen	7
Pin Functions	8
Motorstrom einstellen	9
Traditionelle Variante am Poti des Treibers	9
Einstellung in der Firmware über UART-Mode	10
3. Fysetc Spider Board und die Nutzung mit Klipper per Mainsail Ui	11
Übersicht	11
MainsailOS Image flashen	12
Bootloader auf MCU flashen	12
Firmware compilen	13
Firmware uploaden (DFU)	14
MCU in den DFU-Modus versetzen	14
Firmware uploaden	14
4. printer.cfg	15
LED-Neopixel Steuerung über Makros	16
5. Klipper Konsolen Befehle:	16
Klipper G-Code	16
6. Macros	17
Start/End/Pause/Cancel Macros	17
START-Skript	17
END-Skript	17
PAUSE-Skript	18
CANCEL-Skript	18
Anpassungen in PrusaSlicer:	18
START GCode	18
END GCode	18
Macros für LED-Steuerung	19
Macro für Bed-Mesh-leveling	19
Macro um GCode Fehlermeldungen auszufiltern	19

7. Pin Belegung Spider Board für Config.....	20
8. Belegung Flachbandkabel	22
Informationen zum Flachbandkabel	22
Informationen zum Wannenstecker mit Schneidklemmtechnik	22
9. Mechanik.....	23
Linearlager	23
Gleitlager	23
Linearwellen	23
GT2 Zahnriemen	24
10. Probe - für Motorisiertes automatisches Bett leveling	25
Auswahl der Sensoren	25
Vorhandene Sonden:.....	25
Unterschied NPN / PNP	26
Konfiguration der Probe	27
Funktionstest der Probe	27
Ausrichten der Probe.....	27
Kalibration der Probe	27

TODO:

 schaubild mit probe points / motoren und positionsangaben

 config datei anfügen sobald Drucker druckt

 Deckblatt erstellen

 Marquardt Typ 1050

1. Spider Board V1.0

3.3V Problematik

Vor dem aufstecken der Schrittmotortreiber ist darauf zu achten die 24V kurzzuschließen um den gespeicherten Strom der Kondensatoren zu verbrauchen.

Dies gilt auch nach jedem Tausch der Schrittmotortreiber!

Um die Platine kurzzuschließen, einen 100K Widerstand oder ein Kabel zwischen 24V und GND, wie in den untenstehenden Bildern gezeigt wird. 5 min warten, danach kann der Treiber wieder aufgesteckt werden.

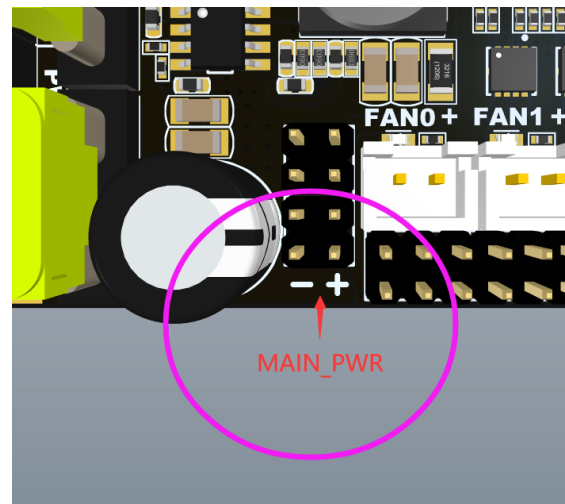
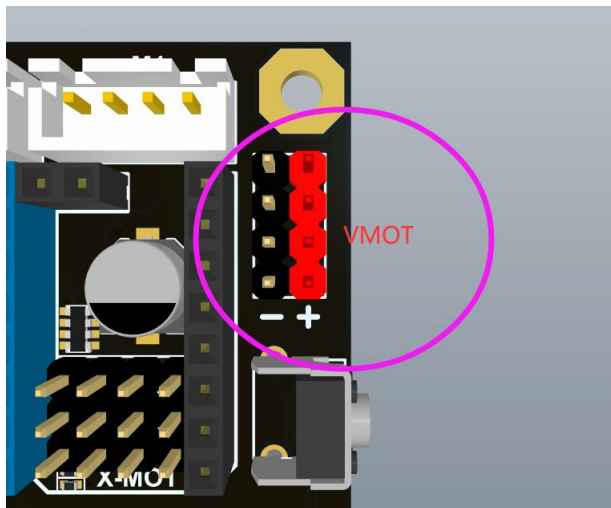
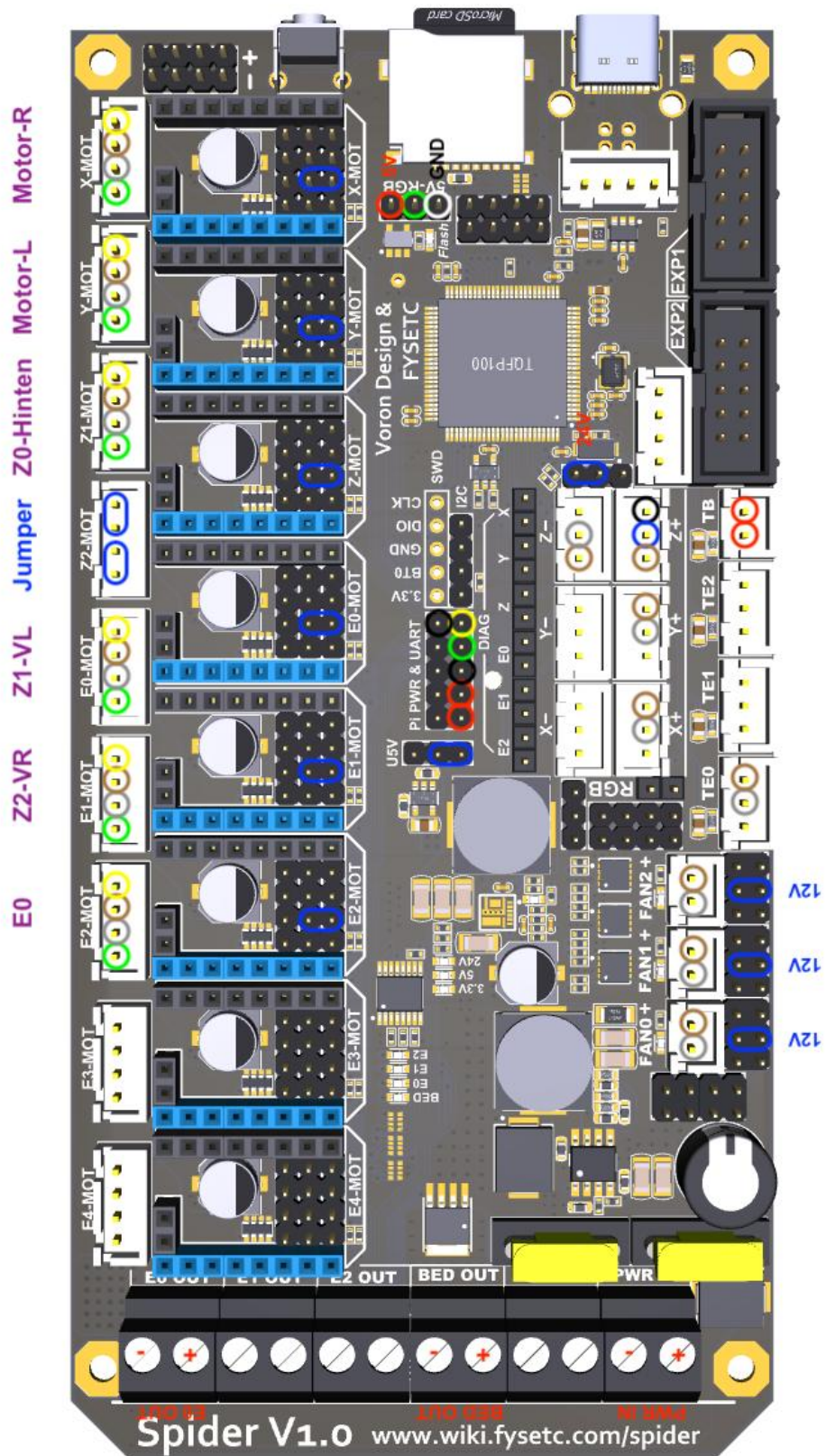
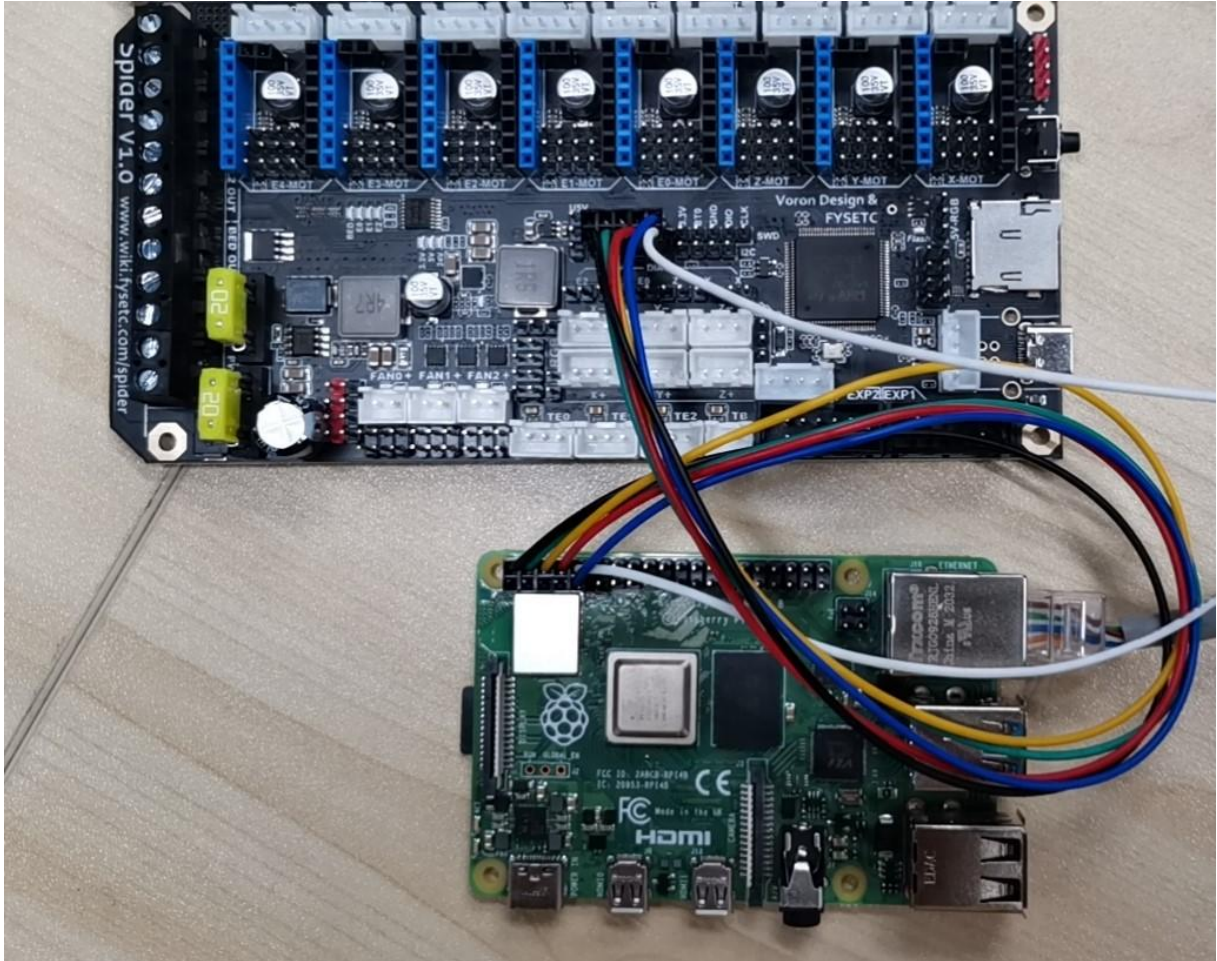


Schaubild Verdrahtung



Hardwareseitige Konfiguration - UART Verbindung Raspberry Pi

Der Raspberry Pi kann über das Spider Board und dem mitgelieferten Kabel mit Strom versorgt werden. Dafür sollte man 3A/5V (15Watt) mit einberechnen.



Auch die UART Verbindung wird mit dem Verbindungskabel ermöglicht und benötigt keine USB Verbindung. Dies muss im Pi, wie im folgenden Abschnitt beschrieben, konfiguriert werden.

Softwareseitige Konfiguration - UART Verbindung Raspberry Pi

1. SSH Verbindung auf Pi herstellen.

```
sudo nano /boot/cmdline.txt
```

2. lösche folgende Passagen:

```
"console=serial0,115200" or "console=ttyAMA0,115200"
```

```
sudo reboot
```

3. erneute SSH Verbindung herstellen

```
sudo raspi-config
```

→Interfacing Option

→ Serial

→NO

→YES

→Ok

→Finish

→Yes

```
sudo reboot
```

4. erneute SSH Verbindung herstellen

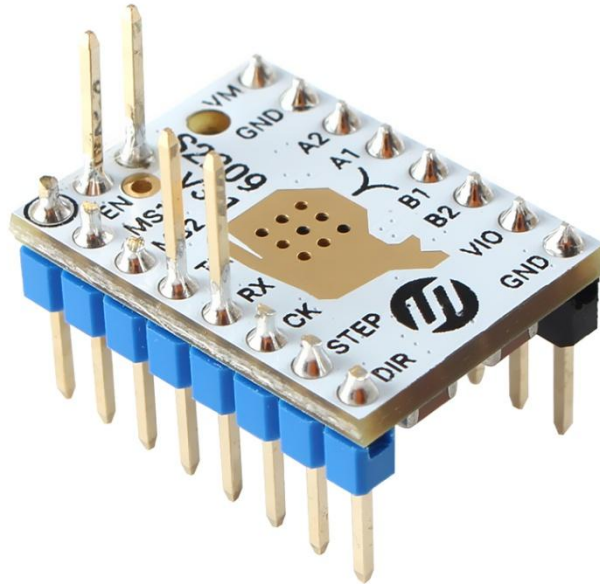
```
sudo nano /boot/config.txt
```

5. Ergänze folgende Zeile am Ende der Datei:

```
dtoverlay=pi3-disable-bt
```

```
sudo reboot
```

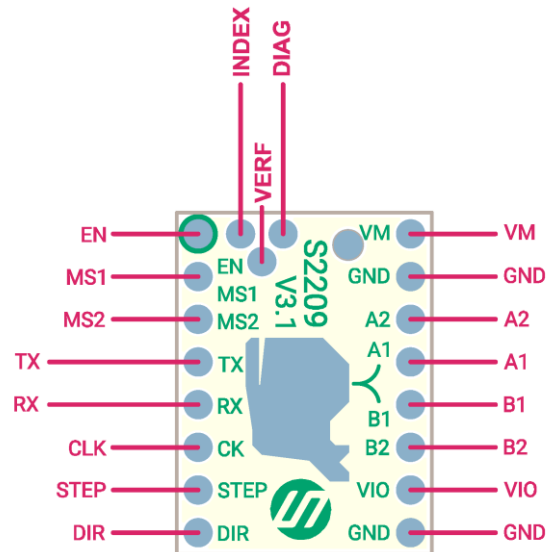

2. TMC 2209 V3.1 Fysetc



Technische Spezifikationen

Model	TMC2209
Interface	Step/Dir
Configuration	CFG Pins or UART
Native Microsteps	up to 1/256
microPlyer Microsteps	1/256
Logic Voltage (VIO)	3-5V
Motor Voltage (VM)	5.5-28V
Motor Phase Current max	2A RMS, 2.8A Peak
Internal V- Regulator	enabled
RDSon	0.1 Ohm (HV 0.2 Ohm)
stealthChop (quiet)	yes
spreadCycle	yes
coolStep	yes
stallGuard	yes
dcStep	yes

Pin Functions



Pin	Function
Power Supply	
GND	Ground
VM	Motor Supply Voltage
VIO	Logic Supply Voltage
Motor Outputs	
M1A	Motor Coil 1
M1B	Motor Coil 1
M2A	Motor Coil 2
M2B	Motor Coil 2
Control Inputs	
STEP	Step-Signal Input
DIR	Direction-Signal Input
TMC2209	
EN	Enable Motor Outputs: GND=on, VIO=off
MS1	Microstep resolution configuration (internal pull-down resistors)
MS2	MS1: 00: ?, 01: ½, 10: ¼ 11: 1/16 For UART based configuration selection of UART Address 0...3
SP	Chopper mode selection: Low/pin open=StealthChop, High=SpreadCycle
CLK	CLK input. Tie to GND using short wire for internal clock or supply external clock.
TX	UART TX, Connected to the PDN via a 1K resistor on board
RX	UART RX, Directly connected to the PDN
DIAG	Diagnostic and StallGuard output. Hi level upon stall detection or driver error. Reset error condition by ENN=high.
INDEX	Configurable index output. Provides index pulse.
VREF	Analog Reference Voltage

Motorstrom einstellen

Es gibt folgende zwei Möglichkeiten den Motorstrom einzustellen.

Traditionelle Variante am Poti des Treibers

Vorsicht!

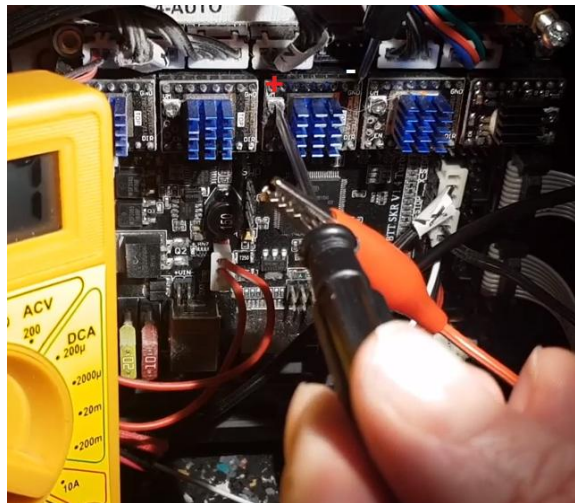
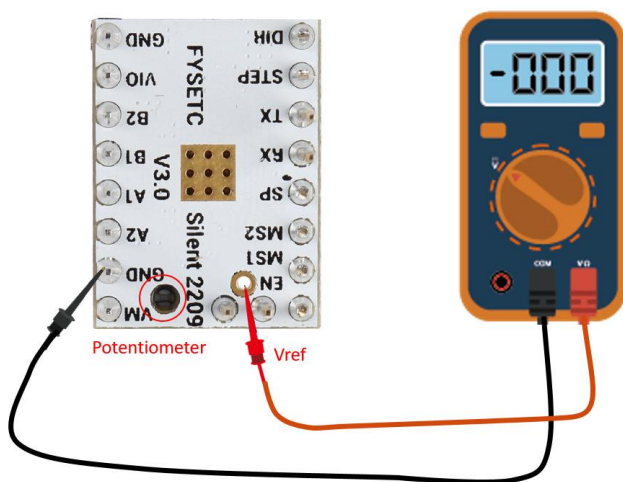
- Niemals den Motor am Treiber angeschlossen lassen.
- Der Treiber muss auf dem mit Strom versorgten Board stecken.
- Messrichtung unbedingt beachten.
- Niemals mehr als 1.2A einstellen.

Am Pin vRef oder am Poti wird der Pluspol angesteckt.

Dafür empfiehlt sich eine Krokodilklemme zwischen Pluspol des Multimeters und Schraubenzieher.

Damit kann man direkt messen und einstellen.

Am Pin GND kann der Minuspol gemessen werden.



Einstellung in der Firmware über UART-Mode

Um den Motorstrom einzustellen:

M906 – Zeigt den Motorstrom der jeweiligen Motoren

M906 X500 – Stellt den Motorstrom der X-Achse auf 0.5A ein

M500 – speichert die Einstellung ins EPROM

Damit die Werte bei einem Firmwareupdate noch vorhanden sind, sollte man den Motorstrom in der Klipper config Datei ergänzen.

Beispielkonfiguration beim X-Motor:

```
#####
# TMC UART configuration
#####
[tmc2209 stepper_x]
uart_pin: PE7
interpolate: true
run_current: 0.8
hold_current: 0.7
sense_resistor: 0.110
stealthchop_threshold: 999999
#####
```

3. Fysetc Spider Board und die Nutzung mit Klipper per Mainsail Ui

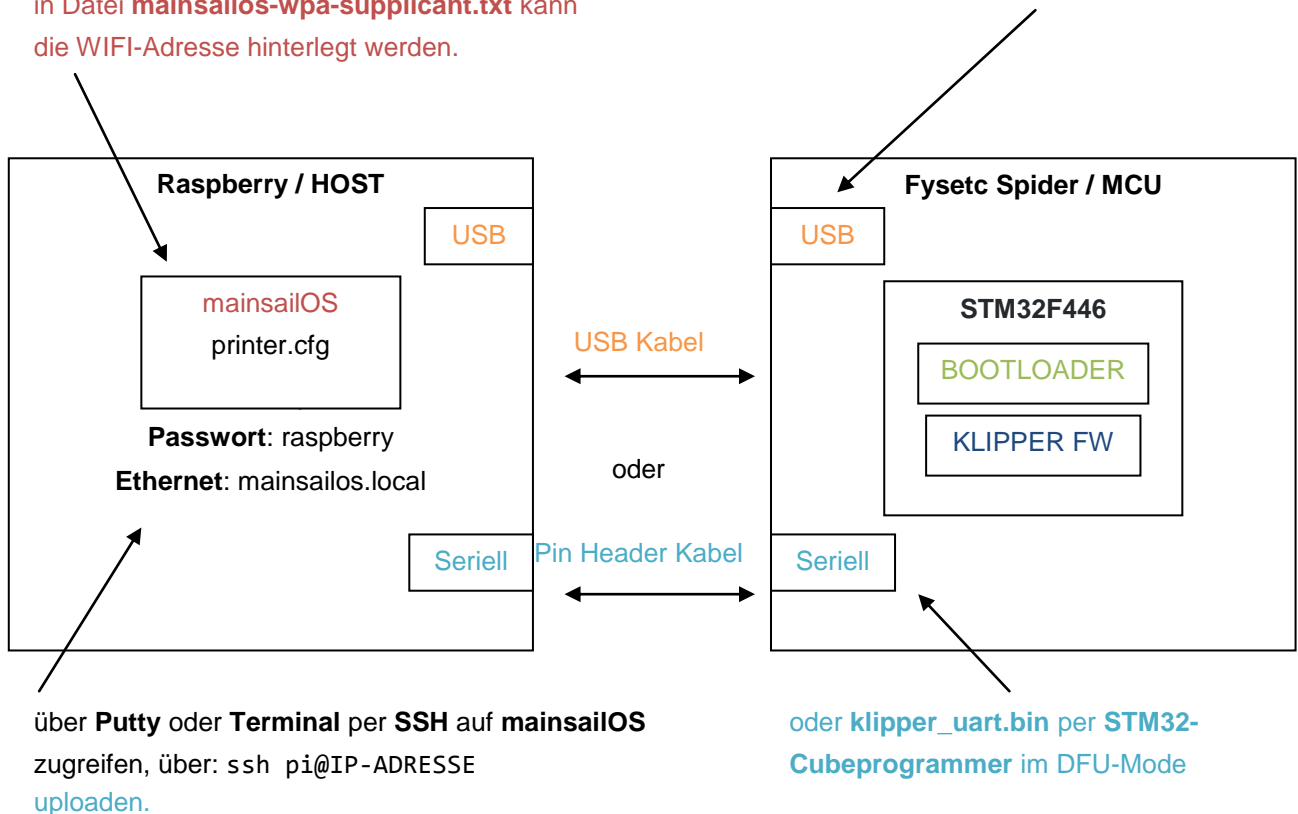
Übersicht

mainsailOS image auf SD-Karte flashen.

printer.cfg auswählen und abändern.

in Datei **mainsailos-wpa-supplicant.txt** kann die WIFI-Adresse hinterlegt werden.

klipper.bin per **STM32-Cubeprogrammer** im DFU-Mode uploaden.



MainsailOS Image flashen

Quellen:

<https://github.com/raymondh2/MainsailOS/releases>

<https://www.klipper3d.org/Installation.html>

<https://github.com/FYSETC/FYSETC-SPIDER>

1. Download latest MainsailOS Image:

<https://github.com/raymondh2/MainsailOS/releases>

2. Dokumentation Installation:

<https://docs.mainsail.xyz/setup/mainsail-os>

3. In folgender Datei auf dem Image kann das WIFI-Setup eingetragen werden:

`mainsailos-wpa-supPLICANT.txt`

4. per "**mainsailos.local**" oder der "**IP-Adresse**" (über Fritzbox herausfinden) öffnet sich nun die Weboberfläche. Hier können Systemupdates sowie für Klipper und Mainsail durchgeführt werden.
5. mittels Putty oder Terminal per SSH mit dem Pi verbinden

`ssh pi@IPADRESSE`

Nutzer: `pi`

Passwort: `raspberry`

Bootloader auf MCU flashen

1. STM32Cubeprogrammer öffnen
2. Beachte bei der Nutzung des STM32Cubeprogrammer alle Schritte wie auf der letzten Seite beschrieben.
3. "**Bootloader_FYSETC_SPIDER.hex**" uploaden.

Link zur Anleitung:

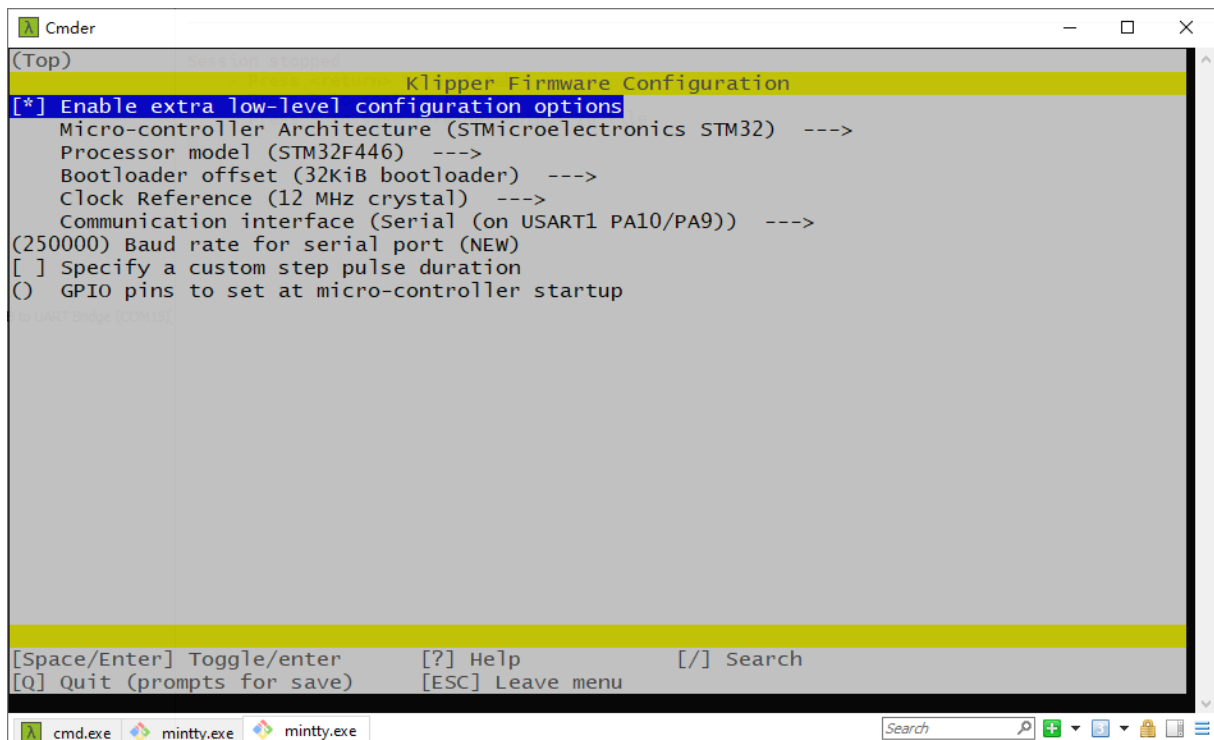
<https://github.com/FYSETC/FYSETC-SPIDER/tree/main/bootloader>

Firmware compilen

```
cd klipper
```

```
make clean
```

```
make menuconfig
```



```
make
```

```
ls-l /dev/serial/by-id/
```

```
sudo service klipper stop
```

```
make flash FLASH_DEVICE=DEVICEPATH
```

```
sudo service klipper start
```

Falls sich das gebaute Makefile nicht über den Pi flashen lässt, am besten mittels **Swish** oder **FileZilla** vom Pi ziehen und per **STM32CubeProgrammer** auf das Spider Board laden.

Das gebaute Makefile liegt im Verzeichnis *klipper/out/klipper.bin*

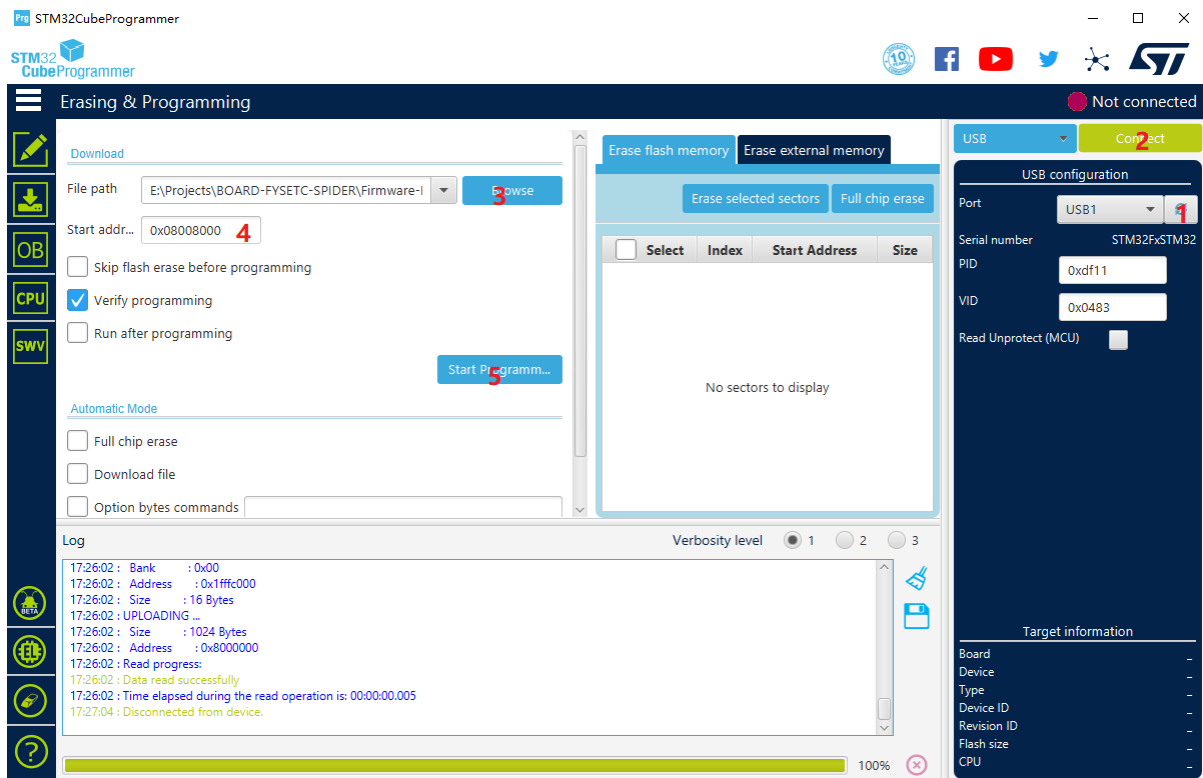
Firmware uploaden (DFU)

Die Software STM32 Cube Programmer wird zum flashen der Firmware verwendet

MCU in den DFU-Modus versetzen

1. Spider Board abstecken (stromlos)
2. BTO zu 3.3V jumpen (Mitte Board)
3. Board per USB am PC anschließen → DFU Mode
4. Jumper abstecken

Firmware uploaden



1. Click the button to find the DFU port.
2. Connect the DFU
3. Choose the "firmware.bin" file.
4. "Start address" muss bei Klipper mit 32k boot address zu **0x8008000** gesetzt werden.
5. Start Programming

4. printer.cfg

Auf der MainsailOS Weboberfläche unter "Einstellungen" muss eine "printer.cfg" erstellt werden.

Diese kann man mit den "config_examples" erstellt werden, bzw. mit:

<https://github.com/FYSETC/FYSETC-SPIDER/blob/main/firmware/Klipper/printer.cfg>

Für die Pinbelegung gilt:

"!" = Reverse

"^" = Eingang mit pull Up Widerstand

"~" = Eingang mit pull Down Widerstand

- Es muss zusätzlich die **mainsail.cfg** mit eingebunden werden, einfach in der printer.cfg folgende Zeile ergänzen:

```
[include mainsail.cfg]
```

- Der MCU **Device Path** muss auch richtig angegeben werden damit der Pi mit dem Spider Board kommunizieren kann:

```
#####
##      Connection: Pi - USB - Spider:
##      upload klipper.bin and use device path as:
serial: /dev/serial/by-id/usb-Klipper_stm32f446xx_230032000851363131363530-if00
#####
##      Connection: Pi - uart0 - Spider:
##      upload klipper_uart.bin and use device path as:
serial: /dev/ttyAMA0
#####
```

- Es ist außerdem ratsam `restart_method: command` mit in die Config einzufügen falls sich die Firmware bei Änderungen nicht von selbst resettet.

LED-Neopixel Steuerung über Makros

```
[neopixel my_neopixel]
pin: PD3
chain_count: 31
color_order: GRB
initial_RED: 0.2
initial_GREEN: 0.3
initial_BLUE: 0.4
#   Sets the initial LED color of the Neopixel. Each value should be
#   between 0.0 and 1.0.
#   LEDs. The default for each color is 0.

[gcode_macro LEDOFF]
gcode: SET_LED LED=my_neopixel RED=0.0 GREEN=0.0 BLUE=0.0 [INDEX=<index>]
[TRANSMIT=0] [SYNC=1]

[gcode_macro WORKLIGHT]
gcode: SET_LED LED=my_neopixel RED=1.0 GREEN=1.0 BLUE=1.0 [INDEX=<index>]
[TRANSMIT=0] [SYNC=1]

[gcode_macro REDLOUNGE]
gcode: SET_LED LED=my_neopixel RED=1.0 GREEN=0.0 BLUE=0.0 [INDEX=<index>]
[TRANSMIT=0] [SYNC=1]
```

5. Klipper Konsolen Befehle:

status – prüft die config Datei
restart – lädt die config Datei neu
help – Liste von Befehlen

Klipper G-Code

<https://github.com/KevinOConnor/klipper/blob/master/docs/G-Codes.md>

6. Macros

Start/End/Pause/Cancel Macros

Folgende Makros können in printer.cfg oder zur besseren Übersicht in einer extra macro.cfg definiert werden. Dies hat den Vorteil dass im Slicer nur noch auf das Makro verwiesen werden muss.

START-Skript

```
[gcode_macro START_PRINT]
gcode:
  {% set EXTRUDER_TEMP = params.EXTRUDER_TEMP|default(0)|float %}
  {% set BED_TEMP = params.BED_TEMP|default(0)|float %}
  M140 S{BED_TEMP}      # Start bed heating
  M104 S{EXTRUDER_TEMP} # Start Nozzle heating
  G90                   # Use absolute coordinates
  SET_GCODE_OFFSET Z=0.0 # Reset the G-Code Z offset (adjust Z offset if needed)
  G28                   # Home the printer
  G1 X190 Y0 Z5 F2000    # Move the nozzle near the bed
  M190 S{BED_TEMP}       # Wait for bed to reach temperature
  M109 S{EXTRUDER_TEMP}  # Set and wait for nozzle to reach temperature
  G1 Z0.2 F300           # Move the nozzle very close to the bed
  G92 E0                 # reset Extruder
  G1 X190 Y0             # Extrudiere von
  G1 X190 E5 F200        # Bis
  G1 X145 E15 F200       # Extrudiert Linie
  G1 X125 F2000          # Wipe
  G92 E0                 # reset Extruder
```

END-Skript

```
[gcode_macro END_PRINT]
gcode:
  M140 S0                # Turn off Bed
  M104 S0                # Turn off Extruder
  M107                   # Turn off fan
  G91                   # relativ
  G1 X-2 Y-2 E-3 F300    # Move nozzle away from print while retracting
  G1 Z10 F2000           # Raise nozzle by 10mm
  G28 X Y Z              # Home
  G90                   # absolut
  M84                   # Disable steppers
```

PAUSE-Skript

```
[pause_resume]

[gcode_macro M600]
gcode:
    {% set X = params.X|default(50)|float %}
    {% set Y = params.Y|default(0)|float %}
    {% set Z = params.Z|default(10)|float %}
    SAVE_GCODE_STATE NAME=M600_state
    PAUSE
    G91
    G1 E-.8 F2700
    G1 Z{Z}
    G90
    G1 X{X} Y{Y} F3000
    G91
    G1 E-50 F1000
    RESTORE_GCODE_STATE NAME=M600_state
```

CANCEL-Skript

```
[gcode_macro CANCEL_PRINT]
gcode:
    M220 S100 ; Reset Speed factor override percentage to default (100%)
    M221 S100 ; Reset Extrude factor override percentage to default (100%)
    M104 S0    # Turn off Extruder
    M140 S0    # Turn off Bed
    M107       # Turn off fan
    G91       # relativ
    G1 X-2 Y-2 E-3 F300 # Move nozzle away from print while retracting
    G1 Z10 F2000       # Raise nozzle by 10mm
    G28 X Y Z          # Home
    G90               # absolut
    M84               # Disable steppers
```

Anpassungen in PrusaSlicer:

START GCode

```
START_PRINT BED_TEMP=[first_layer_bed_temperature]
EXTRUDER_TEMP=[first_layer_temperature]
```

END GCode

```
END_PRINT
```

Macros für LED-Steuerung

```
[gcode_macro LEDOFF]
gcode: SET_LED LED=my_neopixel RED=0.0 GREEN=0.0 BLUE=0.0 [INDEX=<index>]
[TRANSMIT=0] [SYNC=1]

[gcode_macro WORKLIGHT]
gcode: SET_LED LED=my_neopixel RED=1.0 GREEN=1.0 BLUE=1.0 [INDEX=<index>]
[TRANSMIT=0] [SYNC=1]

[gcode_macro REDLOUNGE]
gcode: SET_LED LED=my_neopixel RED=1.0 GREEN=0.0 BLUE=0.0 [INDEX=<index>]
[TRANSMIT=0] [SYNC=1]
```

Macro für Bed-Mesh-leveling

```
[gcode_macro G29]
gcode: BED_MESH_CALIBRATE
```

Macro um GCode Fehlermeldungen auszufiltern

Hierfür wird ein unnötiger Gcode als "Leer" definiert

```
[gcode_macro M10]
gcode:
```

7. Pin Belegung Spider Board für Config

Die Pin Belegung sollte beim laden der Beispiel Printer.cfg für das Spider Board über die Mainsail Ui passen. Hier können die jeweiligen Pins überprüft werden.

X	STEPPER	
STEP	PE11	
DIR	!PE10	
ENABLE	!PE9	
ENDSTOP	^!PA1	PB14
uart_pin	PE7	
Y	STEPPER	
STEP	PD8	
DIR	!PB12	
ENABLE	!PD9	
ENDSTOP	^!PA2	^PB13
uart_pin	PE15	
Z	Hinten	
STEP	PD14	
DIR	PD13	
ENABLE	!PD15	
ENDSTOP	^!PA0	^!PA3
uart_pin	PD10	
Z1	Vorne-Links	
STEP	PD5	
DIR	PD6	
ENABLE	!PD4	
uart_pin	PD7	
Z2	Vorne-Rechts	
STEP	PE6	
DIR	PC13	
ENABLE	!PE5	
uart_pin	PC14	

E0	STEPPER
STEP	PE2
DIR	!PE4
ENABLE	!PE3
uart_pin	PC15
HEATER PIN	PB15
SENSOR PIN	PC0

heater_bed	
heater_pin	PB4
sensor_pin	PC3

FAN0	Bauteillüfter
Pin	PB0

FAN1	Hotendfan
Pin	PB1

FAN2	Controll Board
Pin	PB2

Probe	Induktivsensor
Braun	VPROBE
Blau	GND
Schwarz	^!PA3

Probe muss NPN sein!

8. Belegung Flachbandkabel

24V / AWG26 = 0,14mm²

Motor A max 1,4A	1	2	Motor A max 1,4A
Motor B max 1,4A	3	4	Motor B max 1,4A
max. 40W Heizpatrone +	5	6	Hotendlüfter +
max. 40W Heizpatrone -	7	8	Hotendlüfter -
FREI	9	10	Probe Trigger (SW)
FREI	11	12	Probe + (BN)
FREI	13	14	Probe - (BL)
Bauteillüfter (-)	15	16	Bauteillüfter (+)
Thermistor	17	18	Thermistor
X-Endstop	19	20	X-Endstop

(Sicht von vorne auf den Stecker am Druckkopf)

Informationen zum Flachbandkabel

3M 3801 Serie

20 Pin

AWG 26 (7x0.16)

Rastermaß/Pitch: 1.27mm

Link zum Kauf in Meterware:

<https://de.farnell.com/3m/3801-20/flachbandkabel-pvc-20polig-per/dp/1797568>

AWG	Querschnitt	R(Ω/km)
AWG 30	0,051	349
AWG 28	0,081	441
AWG 26	0,129	556

Informationen zum Wannenstecker mit Schneidklemmtechnik

IDC (Insulation Displacement Connector)

Raster: 0,050" (1,27mm)

20 Pin / 2 Reihen

9. Mechanik

Linearlager

Misumi - LM10UU

Innendurchmesser: 10mm

Toleranz: 0µ / -9µ

perfekt geeignet für **g6** Wellen (Toleranzgrenze in µm: -6 bis -17)

Misumi empfiehlt **Alvania Fett S2** von **Shell**
Lager im Sparklab Shop sind Misumi Lager

Gleitlager

China **Messinggleitlager** mit Graphiteinsätze:

Toleranzklasse in µm: 0 bis -15

Kunststoffgleitlager von Igus benötigen im verbauten Zustand eine Vorspannung, ansonsten haben diese zu viel Spiel. Sie werden nicht empfohlen da die Lager zu sehr gequetscht werden.

Das Losreismoment ist beachtlich größer als das von Linearlagern mit Kugeln.

drylin® R Lineargleitbuchse **RJ4JP-01** (grau/ japanabmessungen)
0 bis +40

drylin® R Lineargleitbuchse **RJMP-01** (gelb)
0 bis +40

Linearwellen

Material: EN 1.3505 Äquivalent

Toleranz: 10mm g6

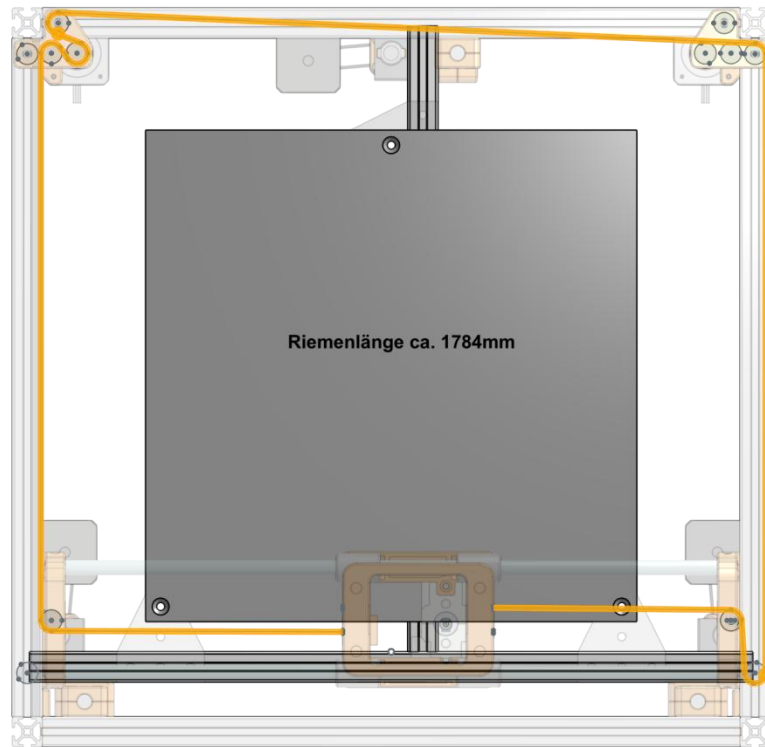
X-Achse 2x Wellen 10mm g6 x 459mm

Y-Achse 2x Wellen 10mm g6 x 433mm

Z-Achse 3x Wellen 10mm g6 x 440mm

X/Y Antrieb 2x Wellen 8mm g6 x 346mm

GT2 Zahnriemen



GT2-2MGT-9 GATES

Definition der Bezeichnung:

Profil: GT2

Zahnabstand: 2mm

Riemenbreite: 9mm

alternativ POWGE GT2 9mm

10. Probe - für Motorisiertes automatisches Bett leveling

Auswahl der Sensoren

Für das Spider Board muss eine **NPN**-Probe benutzt werden.

NC oder NO funktioniert beides, dies kann in der Config angepasst werden.

NC macht mehr Sinn, da ein Kabelbruch sofort entdeckt wird.

Vorhandene Sonden:

Bezeichnung: **IFM**

Art: **induktiv**

Schaltabstand: **2mm**

PNP/NO

Bezeichnung: **di-soric KDCT 08 V 02 G3-T3**

Art: **kapazitiv**

Schaltabstand: **2mm**

NPN/PNP - NO/NC

Bezeichnung: **PL-08N**

Art: **induktiv**

Schaltabstand: **8mm**

NPN/NO

Unterschied NPN / PNP

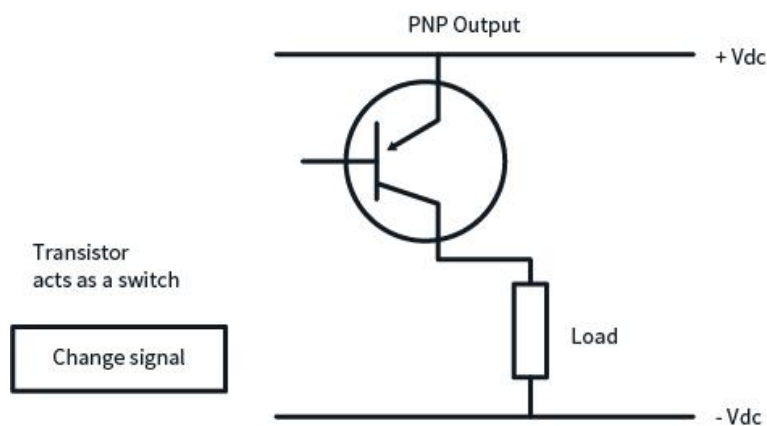
PNP Sensoren werden hauptsächlich in **Europa** eingesetzt.

NPN Sensoren werden noch zum Teil in **Asien** eingesetzt.

PNP-Sensor:

Bei PNP-Sensoren wird die Last mit dem Schaltausgang und V^- verbunden; jetzt ist V^- der Bezugspunkt. Ergibt sich am Sensor ein Signalwechsel, so schaltet der Transistor durch. Der Strom fließt von V^+ durch den Transistor und über die Last zu V^- , wodurch der Stromkreis geschlossen wird.

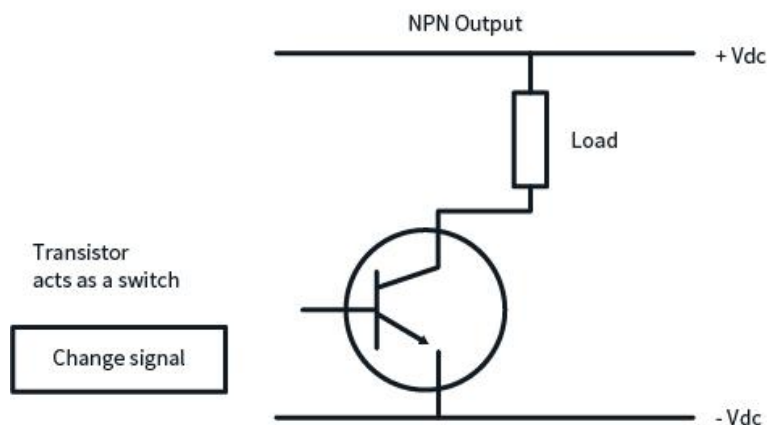
PNP Sourcing current



NPN-Sensor:

Bei NPN-Sensoren wird die Last mit dem Schaltausgang und V^+ verbunden; V^+ ist der Bezugspunkt. Wird am Sensor ein Signalwechsel herbeigeführt, schaltet der Transistor durch, Strom fließt von V^+ über die Last durch den Transistor zu V^- , wodurch der Stromkreis geschlossen wird.

NPN Sinking current



Konfiguration der Probe

Funktionstest der Probe

Zuerst sollte man prüfen ob die Probe richtig funktioniert, um den Zustand zu ermitteln kann folgender Terminalbefehl verwendet werden. Ausgabe "triggered" oder "open".

`QUERY_PROBE`

Falls NO/NC vertauscht ist, kann dies in der config mit `^PA3` oder `^!PA3` geändert werden.

Wenn alles funktioniert kann mit `PROBE` eine einzelne Probe durchgeführt werden.

Ausrichten der Probe

Absolute Positionierung

`G90`

Zur Bettmitte fahren mit

`G1 X170 Y200`

- Probe lösen und Abstandshalter nach jeweiligem Schaltabstand der Probe unterlegen.
- Düse so weit in Richtung Bett fahren bis Papier daran schleift
- Probe festschrauben
- Abstandshalter entfernen
- Falls der Schalterpunkt direkt an der Probe verstellt werden kann, muss der erste Schalterpunkt nun gesetzt werden.
- Düse 1mm vom Bett weg bewegen
- zweiten Schalterpunkt setzen

Kalibration der Probe

`PROBE_CALIBRATE`

Wenn dieser Fehler erscheint:

`Failed to home probe: Timeout during homing`

muss in der config am `[stepper_z]` die `position_min: 0` niedriger geschraubt werden

falls `PROBE_CALIBRATE` funktioniert, Papertest unter Nozzle und mit

`TESTZ Z=-.1`

vorsichtig an das Papier tasten.

Wenn Papier leicht streift kann die Prozedur mit `ACCEPT` abgeschlossen werden

21:42:10	probe: z_offset: -0.100 The SAVE_CONFIG command will update the printer config file with the above and restart the printer.	
21:42:08	accept	
21:41:49	Z position: ?????? --> -4.036 <-- -3.936	

SAVE_CONFIG

speichert die Kalibrierung in der printer.cfg - diese muss wieder in printer.cfg umbenannt werden.

IST Z-Wert auf `position_max` in printer.cfg dazu addieren und auch `position_endstop` mit dem gleichen Wert abändern.

Nun sollte Z-0 eine Papierdicke vom Bett entfernt sein. Wenn der Drucker auf Z-Max homed ist er beispielsweise auf 207.86mm

Nun kann `z_tilt` – der eigentliche MABL Vorgang gestartet werden. Dies kann über die Schaltfläche oder über den Gcode passieren.

Falls die Düse immer noch nicht richtig zum Bett ausgerichtet ist, kann es sein dass die X/Y Achse nicht parallel zum Bett ausgerichtet ist.