

# BD Sensor

- [https://github.com/markniu/Bed\\_Distance\\_sensor/wiki](https://github.com/markniu/Bed_Distance_sensor/wiki)

## YouTube Video #108



## Hinweise

Zu Beginn vielleicht ein paar Eindrücke von mir zu dem Sensor:

- Unordered List Item Wenn er einmal eingestellt ist, dann funktioniert der Sensor gut und auch zuverlässig. Ich finde allerdings die Konfiguration schon ziemlich kompliziert. Mir fehlt da in der original Anleitung auch noch etwas der rote Faden. Das ist bei anderen Eddy Sensoren mitunter deutlich besser gelöst.
- Die Einstellung von second\_homing\_speed fand ich sehr zeitaufwendig und fehlerträchtig. Hier kann man sich schnell das Druckbett ruinieren (bei mir passiert) oder den Kopf verbiegen. Also hier sehr bedacht vorgehen und immer einen Finger am Ausschalter ...
- Die Halterung für den StealthBurner mag ich persönlich gar nicht. Die ist in meinen Augen eine Fehlkonstruktion. Man kriegt das nur mit großen Mühen korrekt eingestellt (oder ich hab zu

linke Hände 😊 ).

- Der Umstand das der Sensor I2C nutzt gefällt mir auch nur so semi. Ich denke das macht es für viele User sehr kompliziert weil beim Klipper Kompilieren für das Druckerboard immer an den BD Sensor Code gedacht werden muss. Für einen Nichtentwickler mitunter etwas verwirrend.
- Mit meinem Spider V1 Board habe ich den Sensor nicht ans Laufen bekommen. Hier kamen immer I2C Fehler (Siehe ganz unten auf dieser Seite). Ich könnte das nur durch einen Board Tausch auf ein Spider H7 lösen.
- Die Anleitung hier erhebt keinen Anspruch auf Vollständigkeit. Der ganze BDSensor Umbau hat bei mir aus diversen Gründen sicherlich 3 Wochen gedauert. Gut möglich das ich hier nicht mehr alles niederschreiben konnte. Aber ich hoffe das Groh notiert zu haben.
- Ich hatte auch Probleme mit der Black Print Druckplatte. Oder sagen wir generell mit Druckplatten die eine dickere nicht metallische Oberfläche haben. Der Sensor kam bei mir damit nur selten korrekt zurecht. Erst eine normale strukturierte PEI Platte hat da Abhilfe geschaffen.
- Grundsätzlich sollte man sich vorher nochmal genau diese Seiten der original Doku ansehen: [https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Installing-for-Klipper](https://github.com/markniu/Bed_Distance_sensor/wiki/Installing-for-Klipper)

[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Collision-sensing](https://github.com/markniu/Bed_Distance_sensor/wiki/Collision-sensing)  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Homing-with-contactless-and-contact-probing-together](https://github.com/markniu/Bed_Distance_sensor/wiki/Homing-with-contactless-and-contact-probing-together)  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Z-move-or-not-automatically\\_Z\\_Tilt\\_QGL](https://github.com/markniu/Bed_Distance_sensor/wiki/Z-move-or-not-automatically_Z_Tilt_QGL)

## Übersicht

- [https://en.wikipedia.org/wiki/Eddy\\_current](https://en.wikipedia.org/wiki/Eddy_current)
- 2 Versionen
  - BDsensor → Sensor & Elektronik im Bauraum
  - BDsensorM → nur Probe im Bauraum und Elektronik steckbar auf EXT1 Port
- kein eigenes USB / CAN Interface !
- Nutzt I2C für die Kommunikation
  - Der Sensor funktioniert mit einem Software I2C Port
- Es muss eine Erweiterung für Klipper installiert werden!

## Anschluss am Board

- Der Sensor muss an einem bestehenden Druckerboard per I2C angeschlossen werden.
- Es werden zwei Pins für I2C benötigt: SDA und CLK und zwei für **5V** und GND
- I2C muss nicht zwingend ein Hardware I2C sein. Die sind oft eh schwer auszumachen. Es reichen 2 Pins auf denen dann ein Software basierter I2C läuft. Die Umsetzung macht Klipper

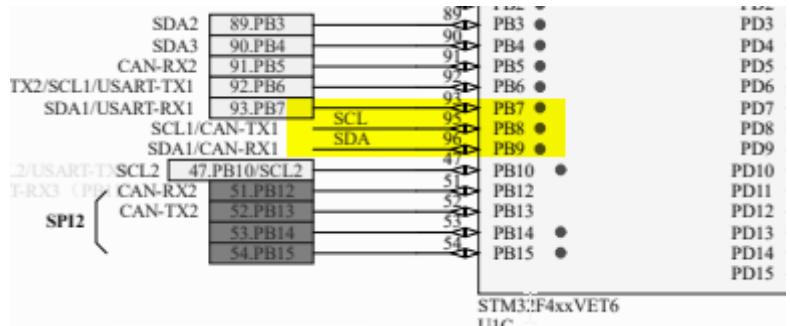
kleine



- Wenn der Sensor Real time Leveling machen soll muss er an dem Board angeschlossen sein, an dem auch die Z motoren angeschlossen sind. Siehe [https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/How-to-run-Real-Time-Level](https://github.com/markniu/Bed_Distance_sensor/wiki/How-to-run-Real-Time-Level)
- Als Pins kann man u.A. am kopfboard einen BLTouch Anschluss verwenden. Der hat in der Regel auch gleich 5V und GND. An einem normalen Druckerboard kann man meist einen EXT1/ETX2 Anschluss verwenden. Generell müssen es Pins sein, die nicht schon durch irgendwelche Logik oder Filter beschaltet sind - also reine IO Pins!

Beispiel für den Anschluss am BLTouch Port:

BLtouch		BDsensor
GND	-->	GND
5V	-->	5V
S	-->	CLK/SCL (Input)
GND	-->	GND
Zmin	-->	SDA (Input/Output)



Beispiel für den Anschluss am Spider V1.0:

Achtung beim BD Sensor M. Hier sind teilweise die Beschriftungen auf der Platine falsch:  
Pins sind 2 Pins verschoben für Clock und SDA

## Klipper kompilieren

- die BLSensor Repo klonen
  - cd ~
  - git clone [https://github.com/markniu/Bed\\_Distance\\_sensor.git](https://github.com/markniu/Bed_Distance_sensor.git)
- BLSensor Addons installieren  
`~/Bed_Distance_sensor/klipper/install_BDsensor.sh`
- Klipper kompilieren
  - cd ~/klipper/
  - make menuconfig
 

**Hinweis** Hier müsst ihr euer Board ganz normal einstellen. Es gibt keine direkte Option für BDsensor!
  - ./make\_with\_bdsensor.sh
  - make flash
 

**Hinweis** Wie ihr genau euer Board flashen müsst hängt vom Board und der Flashmethode ab! make flash steht hier nur als Beispiel und ihr müsst das zwingend an euer Board bzw. eure Verbindung anpassen!
  - Darauf achten das während dem Flashen folgendes erscheint : Compiling `out/src/BD_sensor.o` während man `make_with_bdsensor.sh` ausführt
- Updates kann man auch machen - bedarf aber dann immer einem Klipper Update auf dem Board! eine kurze Beschreibung dazu findet ihr hier:  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Installing-for-Klipper](https://github.com/markniu/Bed_Distance_sensor/wiki/Installing-for-Klipper)

## erster Test

- sicherstellen das [probe] und oder [bltouch] in eurer Konfig nicht mehr aktiv sind
- am besten eine eigene Konfig Datei anlegen für den Sensor → `BDsensor.cfg`
- für einen ersten Test reicht dann folgender Inhalt:

```
[BDsensor]
# Don't use aliases for the board pins
sda_pin: PB1 # example of connecting to main board Creality V4.2.7
scl_pin: PB0
#scl_pin:MKS_THR:gpio20 # example of connecting to CAN module like MKS
THR42
#sda_pin:MKS_THR:gpio11
```

```
#scl_pin:host:gpio17 # example of connecting to GPIO on RaspberryPi
#sda_pin:host:gpio27
delay: 20 # you can set it 10 if the BDsensor version is >=1.2
z_offset:0 # within -0.6 to 0.6mm
x_offset: -34
y_offset: 0
no_stop_probe: # fast probe that the toolhead will not stop at the
probe point, disable it by commenting out.
position_endstop: 1.2 #the triggered position, recommend value is 1~2.8
collision_homing:0 # set it 1 to enable homing with nozzle collision
sensing.
collision_calibrate:0 # set it 1 to enable auto calibrate BDsensor with
nozzle collision sensing.
#QGL_Tilt_Probe:0 #set 1 to enable probe up and down when do
quad_gantry_level
```

**WICHTIG** Ihr müsst hier die I2C Pins korrekt eintragen! Also sda\_pin und scl\_pin anpassen. Das hängt natürlich von eurem Board ab.

- Wenn ihr jetzt in der MainSail Konsole M102 S-1 ausführt müsste folgendes Ergebnis kommen:

```
BDsensorVer:V1.2c pandapi3d
,switch_mode=1,collision_homing=0,collision_cal=0
```

- Wenn der Sensor nicht korrekt angeschlossen oder klipper falsch kompiliert wurde, dann bekommt ihr folgendes:

```
No data or corrupt data from BDsensor(XXXXXXXXXXXXXXXXXXXX), Please
check connection
```

## Einbau

Der Einbau (bei mir im Stealthburner) ist kein größeres Problem. Er wird wie eine normale Probe eingebaut. Schaut dazu am besten das Video an (ab Minute 13:58).

Was mir an der Halterung gar nicht gefällt ist, dass man den Stelathburner fast komplett demontieren muss um den Sensor in der Höhe einzustellen. Extrem unpraktisch und zeitraubend. Zumal man im eingebautem Zustand auch kaum Möglichkeiten bzw. Referenzen hat an denen man etwas nachmessen könnte.

Ich habe deshalb den Kopf immer auf 0,5mm ca. Richtung Bett gefahren (Nozzle zu Bett Abstand) und dann vorsichtig den Kopf demontiert (im bestromten Zustand). Das ist keine wirklich gute Lösung, aber so bleibt wenigstens der Kopf an der richtigen Stelle stehen.

## Konfiguration

## BDSensor.cfg

Ich habe mir für meine Konfiguration eine extra CFG Datei angelegt mit folgendem Inhalt:

### BDSensor.cfg

```
[BDsensor]
# Don't use aliases for the board pins
sda_pin : PD1
scl_pin : PD0
#scl_pin : MKS_THR:gpio20 # example of connecting to CAN
module like MKS THR42
#sda_pin : MKS_THR:gpio11
#scl_pin : host:gpio17 # example of connecting to GPIO on
RaspberryPi
#sda_pin : host:gpio27

# This speed is only for the z probe speed while doing z_tilt. not for
the homing speed
speed : 3
# This is the time period (speed)for the communication between mcu and
bdsensor, that means lower value the higher communication speed.
delay : 20 # you can set it 10 if the BDsensor version is
>=1.2
# This value will be ingnored if nozzlecollision is enabled
z_offset : 0 # within -0.6 to 0.6mm
x_offset : 0
y_offset : 20.0
no_stop_probe: # fast probe that the toolhead will not stop at
the probe point,disable it by commenting out.
# it will stop the z axis if it measure the distance is <=1.2mm while
homing z to avoid crash into the bed. and then set the z value to the
actual read distance value.
# it will not raise nozzle 1.2mm
position_endstop : 1.2 # the triggered position, recommend value is
1~2.8
## The Sensor is attached to the board where the Z stepper are
attached, too.
collision_homing : 0 # set it 1 to enable homing with nozzle
collision sensing.
collision_calibrate : 0 # set it 1 to enable auto calibrate BDsensor
with nozzle collision sensing.
QGL_Tilt_Probe : 1 # set 1 to enable probe up and down when do
quad_gantry_level

#2. If I don't use the Auto Calibration ... Do I need to set an Z
offset so that the first layer can be printed without the nozzle
touching the bed directly?
#Or is it position_endstop?
#### the only difference is that it will home z first with nozzle
```

```
touching to find the 0 position if collision_calibrate:1

#If I set second-homing-speed to high (> 1.5) my stealthburner starts to
bend because the head stops to late.
##### this depends on the mount height and the material of the bed
plate,
#https://github.com/markniu/Bed_Distance_sensor/issues/91#issuecomment-
1966365247

#
https://github.com/markniu/Bed_Distance_sensor/wiki/Z-move-or-not-autom
atically_Z_Tilt_QGL
[gcode_macro QUAD_GANTRY_LEVEL]
rename_existing: _QUAD_GANTRY_LEVEL
description:
gcode:
    #run z tilt with z move up and down at first
    BDSENSOR_SET QGL_TILT_PROBE=1 #set this 1 to enable z axis up and
down
    BDSENSOR_SET COLLISION_HOMING=0
    _QUAD_GANTRY_LEVEL horizontal_move_z=6 retry_tolerance=1
    G28 Z0
    #run z tilt with no z move up and down
    BDSENSOR_SET QGL_TILT_PROBE=0 #set this 0 to disable z axis up and
down while probe
    _QUAD_GANTRY_LEVEL horizontal_move_z=1 retry_tolerance=0.04
    #BDSENSOR_SET COLLISION_HOMING=1 #optional
    G28 Z0

#
https://github.com/markniu/Bed_Distance_sensor/wiki/Homing-with-contact
less-and-contact-probing-together
[gcode_macro G28]
rename_existing: G990028
gcode:
    {%
        if rawparams|length == 0 %}
        { action_respond_info("homing all") }
    G990028 X Y
    ## Homing with contactless probe with the homing_speed in
[stepper_z]
    BDSENSOR_SET COLLISION_HOMING=0
    G990028 Z
    ## Homing with contact probe with the second_homing_speed in
[stepper_z]
    BDSENSOR_SET COLLISION_HOMING=1
    G990028 Z
    {%
        else %}
        {%
            if 'x' in rawparams or 'X' in rawparams %}
            { action_respond_info("homing x") }
            G990028 X
        {%
            endif %}
    {%
        endif %}
```

```

        {%
        if 'y' in rawparams or 'Y' in rawparams  %}
            { action_respond_info("homing y") }
            G990028 Y
        {%
        endif %}
        {%
        if 'z' in rawparams or 'Z' in rawparams  %}
            { action_respond_info("homing z contactless") }
            ## Homing with contactless probe with the homing_speed
in [stepper_z]
            BDSENSOR_SET COLLISION_HOMING=0
            G990028 Z
            { action_respond_info("homing z collision") }
            ## Homing with contact probe with the
second_homing_speed in [stepper_z]
            BDSENSOR_SET COLLISION_HOMING=1
            G990028 Z
        {%
        endif %}
    {%
    endif %}

```

Diese Konfig beinhaltet einige Anpassungen

- Der Versuch ein paar mehr Erklärungen zu den Konfig Parametern einzubauen
- G28 Erweiterung die verhindert das der erste Prob Versuch auf dem Bett erfolgt. Siehe dazu auch  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Homing-with-contactless-and-contact-probing-together](https://github.com/markniu/Bed_Distance_sensor/wiki/Homing-with-contactless-and-contact-probing-together)
- Anpassung für **Quad Gantry Level**. Wenn ihr Z\_TILT verwendet gibt es hier eine Alternative:  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Z-move-or-not-automatically\\_Z\\_Tilt\\_QGL](https://github.com/markniu/Bed_Distance_sensor/wiki/Z-move-or-not-automatically_Z_Tilt_QGL)
- Der **y\_offset** muss korrekt eingestellt werden. Das ist der Abstand von der Nozzle Mitte zur BD Sensor Mitte.
- Wenn ihr Auto Z Probing haben wollt, dann müssen diese beiden Parameter auf 1:  
 collision\_homing & collision\_calibrate. Für den Start und die erste Kalibrierung sollte das aber erstmal auf 0 stehen.

## [safe\_z\_home]

In der Konfig sollte es ein safe\_z\_home geben. home\_xy\_position auf die Bettmitte setzen

```

[safe_z_home]
# https://www.klipper3d.org/Config_Reference.html#safe_z_home
home_xy_position      : 150,150
speed                 : 300
z_hop                 : 5
z_hop_speed           : 10.0

```

## [bed\_mesh]

bed\_mesh muss auch angepasst werden. Hier vor allem die Anzahl der Messpunkte:

[download](#)

```
[bed_mesh]
## https://www.klipper3d.org/Config_Reference.html#bed_mesh
## https://docs.vorondesign.com/tuning/secondary_printer_tuning.html
speed : 200
horizontal_move_z : 1
mesh_min : 10,33
mesh_max : 290,280
probe_count : 9,9
mesh_pps : 3,3
algorithm : bicubic
bicubic_tension : 0.2
## relative_reference_index = (x points * y points) - 1) / 2
## Samples : 3x3 mesh = 4, 5x5 mesh = 12, 7x7 mesh = 24, etc
#relative_reference_index : 12
zero_reference_position : 150, 150
```

Zudem muss zero\_reference\_position auf der Bettmitte stehen und mesh\_min / mesh\_max muss so eingestellt sein das der Sensor über eurem Bett ist!

## **[stepper\_z]**

Hier müssen einige Werte angepasst werden:

[download](#)

```
endstop_pin : probe:z_virtual_endstop
#position_endstop : -0.5
position_max : 290
position_min : -5
# BD Sensor
homing_speed : 8 # 5,8 für Collision Detect
second_homing_speed : 1.4 # 1, 1.4 #set this to 3 if homing with
collision
homing_retract_speed : 2
homing_retract_dist : 0 # 0(Auto Level) / 5 Normal Level
```

- Der Endstop Pin muss auf einen virtuellen Endstop gesetzt werden.
- position\_endstop auskommentieren
- second\_homing\_speed muss man ermitteln wenn man collision Sensing machen möchte. Siehe dazu auch das Video !
- homing\_retract\_dist muss auf 0 auf Collision Sensing (Auto Z Level) und auf 5 bei manuellem Leveln

## Eintrag für Kalibrierung

Bei manueller Kalibrierung muss dieser Eintrag noch in die Konfig:

```
# BD Sensor
[force_move]
enable_force_move : true # required by the command
SET_KINEMATIC_POSITION in the calibration step below.
```

Der Eintrag sollte nach der Kalibrierung wieder entfernt werden.

## Kalibrierung Manuell

- Nozzle reinigen !
- enable\_force\_move aktiviert !
- Gantry ist einigermaßen gerade ausgerichtet
- SET\_KINEMATIC\_POSITION Z=200 in der MainSail Konsole eingeben  
Damit kann der Kopf frei bewegt werden ohne das vorher die Endstops getriggert haben müssen. **Also vorsichtig den Kopf bewegen !!**
- Es ist keine dumme Idee den Kopf erstmal in die Mitte vom Bett zu fahren. Also in etwa so:  
**GO X150 Y150**
- M102 S-6 in der MainSail Konsole eingeben
- Wenn der Durchgang fertig ist mal ein M102 S-5 absetzen. Der erste Wert sollte unter 400 liegen. Ist er darüber, muss der Sensor tiefer! Ein empfohlener Wert wäre ca. 100. Der Folgewert muss dann auf jeden Fall größer sein (um mindestens 10 Punkte).

## Endstop testen

- M119 absetzen und das Ergebnis sollte sein  
Recv: x:open y:open z:open
- Was metallisches unter den Sensor halten und nochmal M119  
Recv: x:open y:open z:TRIGGERED
- Wenn das passt, dann können jetzt alle Achsen gehomt werden mit G28

## Kalibrierung Auto

- Original Anleitung beachten  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Collision-sensing](https://github.com/markniu/Bed_Distance_sensor/wiki/Collision-sensing)
- collision\_homing and collision\_calibrate am 1 setzen
- homing\_retract\_dist auf 0 setzen  
Aber nur wenn man das G28 Kommando ersetzt hat (siehe oben). Ist das nicht ersetzt muss der Wert auf 5.
- zero\_reference\_position bei bed\_mesh auf Bettmitte setzen
- Test mit G28 Z

## Bed Mesh

Wenn alles eingestellt ist, kann man ganz normal ein BED\_MESH\_CALIBRATE starten. Der Sensor fährt dann in Bahnen über das Bett. Das Handling ist genauso wie vorher !

## Firmware BD-Sensor

Die Firmware vom BDSensor kann man bei Bedarf neu Flashen. Dazu gibt es auf der Wiki Seite passende Anleitungen:

- Für Hardware Version 1.1  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Flash-BDsensor\(V1.1\)-firmware](https://github.com/markniu/Bed_Distance_sensor/wiki/Flash-BDsensor(V1.1)-firmware)
- Für Hardware Version >= V1.2  
[https://github.com/markniu/Bed\\_Distance\\_sensor/wiki/Flash-BDsensor\(stm32\)-firmware](https://github.com/markniu/Bed_Distance_sensor/wiki/Flash-BDsensor(stm32)-firmware)

## Hardware

- X Carriage Voron Default  
[https://github.com/VoronDesign/Voron-2/tree/Voron2.4/STLs/Gantry/X\\_Axis/X\\_Carriage](https://github.com/VoronDesign/Voron-2/tree/Voron2.4/STLs/Gantry/X_Axis/X_Carriage)
- x Carriage Voron mit X Endstop  
<https://mods.vorondesign.com/details/snBnPF4Mo9nfFMZdahC9Ng>
- Voron Trident officially uses a slightly improved version of this mod. It is suggested to use the X axis parts from Trident instead  
[https://github.com/VoronDesign/Voron-Trident/tree/main/STLs/Gantry/X\\_Axis](https://github.com/VoronDesign/Voron-Trident/tree/main/STLs/Gantry/X_Axis)
- Lazy BD-SENSOR adapter For Stealthburner Voron  
<https://www.printables.com/model/831679-lazy-bd-sensor-adapter-for-stealthburner-voron>

## Links

- <https://www.pandapi3d.com/bdsensor>
- [https://github.com/markniu/Bed\\_Distance\\_sensor](https://github.com/markniu/Bed_Distance_sensor)
- Klipper Doku - Eddy Current Inductive probe  
[https://www.klipper3d.org/Eddy\\_Probe.html](https://www.klipper3d.org/Eddy_Probe.html)

## ERROR Unknown command: I2CBD

MCU Protocol error

This is frequently caused by running an older version of the firmware on the MCU(s). Fix by recompiling and flashing the firmware.

Your Klipper version is: v0.12.0-349-ga34034494-dirty

MCU(s) which should be updated:

mcu: Current version v0.12.0-349-ga34034494

RasPi: Current version v0.12.0-349-ga34034494

## EBBCan: Current ve

Up-to-

Once the underlying issue is corrected, use the "RESTART" command to bring the system back online.

Architectural Engineering in EGAPP

## NOTES



Startmakro mit 150 Grad !!! → Fila vorweichen

Nozzle Clean → Umbauen !

From:  
<https://drklipper.de/> - **Dr. Klipper Wiki**



Permanent link:  
[https://drklipper.de/doku.php?id=klipper\\_faq:eddy:bd\\_sensor](https://drklipper.de/doku.php?id=klipper_faq:eddy:bd_sensor)

Last update: **2025/01/12 16:20**