BD Sensor

https://github.com/markniu/Bed_Distance_sensor/wiki

YouTube Video #108



Übersicht

- https://en.wikipedia.org/wiki/Eddy_current
- 2 Versionen
 - $\,\circ\,$ BDsensor \rightarrow Sensor & Elektronik im Bauraum
 - \circ 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



- 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
- 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:

Last update: 2025/01/12 16:05 klipper_faq:eddy:bd_sensor https://www.drklipper.de/doku.php?id=klipper_faq:eddy:bd_sensor&rev=1736694352

BLtouch	1	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
 - \circ cd ~
 - o git clone https://github.com/markniu/Bed_Distance_sensor.git
- BLSensor Addons installieren
 - ~/Bed_Distance_sensor/klipper/install_BDsensor.sh
- Klipper kompilieren
 - o cd ~/klipper/
 - make menuconfig

Hinweis Hier müsst ihr euer Board ganz normal einstellen. Es gibt keine direkte Option für BDSensor!

- o ./make_with_bdsensor.sh
- \circ 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/Rod_Distance_concor/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 \rightarrow 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:gpiol1 #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(<u>[[]]</u>[]][][][][][]][]]]), 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 eingbautem 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.
delav
                    : 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</pre>
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 O position if collision calibrate:1 #If I set second-homing-speed to high (> 1.5) my stealtburner 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 qcode: {% 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 %}
            {% 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-pro bing-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
- 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
```

- position_endstop auskommentieren
- second_homing_speed muss man ermitteln wenn man collision Sensing machen möchte. Siehe dazu auch das Video !

[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
                         : 200
speed
horizontal_move_z
                         : 1
mesh min
                         : 10,33
mesh max
                         : 290,280
probe count
                         : 9,9
                         : 3,3
mesh pps
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
<pre>#position_endstop</pre>	: -0.5
position_max	: 290
position_min	: -5
# BD Sensor	
homing_speed	: 8 # 5,8 für Collision Detect
<pre>second_homing_speed</pre>	: 1.4 # 1, 1.4 #set this to 3 if homing with
collision	
<pre>homing_retract_speed</pre>	: 2
homing_retract_dist	: 0 # 0(Auto Level) / 5 Normal Level

 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: **G0 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
- collision_homing and collision_calibrate am 1 setzen

Bed Mesh

Nozzle Collision Sensing

- https://github.com/markniu/Bed_Distance_sensor/wiki/Collision-sensing
- WICHTIG → Retract auf 0 !!!!! https://github.com/markniu/Bed_Distance_sensor/wiki/Homing-with-contactless-and-contact-pro bing-together

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

Hardware

- X Carriage Voron Default 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
- 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
- Klipper Doku Eddy Current Inductive probe 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

Last update: 2025/01/12 16:05 klipper_faq:eddy:bd_sensor https://www.drklipper.de/doku.php?id=klipper_faq:eddy:bd_sensor&rev=1736694352

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 version v0.12.0-349-ga34034494 Up-to-date MCU(s): <none>

Once the underlying issue is corrected, use the "RESTART" command to reload the config and restart the host software.

mcu 'mcu': Unknown command: I2CBD

NOTES

- Kein Collision aktiv vor Kalibrieren ! Kopf macht sonst bei M102 S-6 ein homing und fährt u.Uins Bett !
- Collision Sensing
 - Raw data:467 at 0 mm, BDsensor to bed: 1.8680 mm, Bed: 42.2
 → Kopf zu nah
 - $\circ\,$ Homing Speed max auf 2 eher 1 \rightarrow Sonst setzt die Nozzle auf !
 - warning: triggered in air, 741
- horizontal_move_z : 10
 - \circ max 1-2
 - \circ Messbereich Spule Bed → max 3,5mm
- Bed Mesh Save Meldung auf Screen

11/11

BD Sensor

I Files	KlipperScreen / screen.py	lop ← All Symbols
ਿ master + Q	Code Blame Executable File · 1271 lines (1148 loc) · 52.1 KB	pathlib
	791 def _websocket_callback(self, action, data):	13 References
Q Go to file	840 if selfcur_panels[-1] != "temperature":	
notifications.py	<pre>841 self.show_panel("temperature", extra=self.printer.get_stat("toolhead",</pre>	"extr In this file
	<pre>842 self.show_popup_message(_("Temperature too low to extrude"))</pre>	10 import pa
[] pins.py	843 return	40 - asthlib
power.py	844 elif data.startswith("!! "):	40 = pathilo
0	845 self.show_popup_message(data[3:], 3, from_ws=True)	494 = pathli
pressure_advance.py	846 elif (514 theme cs
printer select.pv	847 "unknown" in data.lower()	Site cheme_es
-	848 and "TESTZ" not in data	> panels/spo
🗋 retraction.py	849 and "MEASURE_AXES_NOISE" not in data	
Settings.pv Setting Settings.pv Settings.pv Settings.pv Settings.	850 and "ACCELERUMEIER_QUERY" not in data	ks_includes
	852 (alf show nonun messame(data from ws-True)	> ks include
🗋 shutdown.py	853 elif "SAVE CONFIG" in data and self printer state == "ready":	- Ro_merade.
splash screen.pv	854 script = {"script": "SAVE CONFIG"}	Q Search for
	855 selfconfirm_send_action(
💾 spoolman.py	856 None,	
Svstem.pv	<pre>857 _("Save configuration?") + "\n\n" + _("Klipper will reboot"),</pre>	
-	858 "printer.gcode.script",	
🕒 temperature.py	859 script	
D updater.pv	860)	
_	861 self.process_update(action, data)	
🗋 zcalibrate.py	862	
scripts	<pre>863 ✓ def process_action(self, action):</pre>	
benpto	864 if action.startswith("prompt"):	
styles	<pre>865 if action.startswith("prompt_begin"):</pre>	
C gitignore	866 if self.prompt is not None:	
	867 self.prompt.end()	
🗋 .readthedocs.yaml	868 self.prompt = Prompt(self)	
	270 return	
	871 self prompt decode(action)	
LICENSE	872 if action startswith("ks show").	
C README md	873 self.parse ks action(action[8:1.strip())	
NLADIVIE.IIIU	and the set of a construction of the set of	

Startmakro mit 150 Grad !!! → Fila vorweichen

Nozzle Clean \rightarrow Umbauen !

