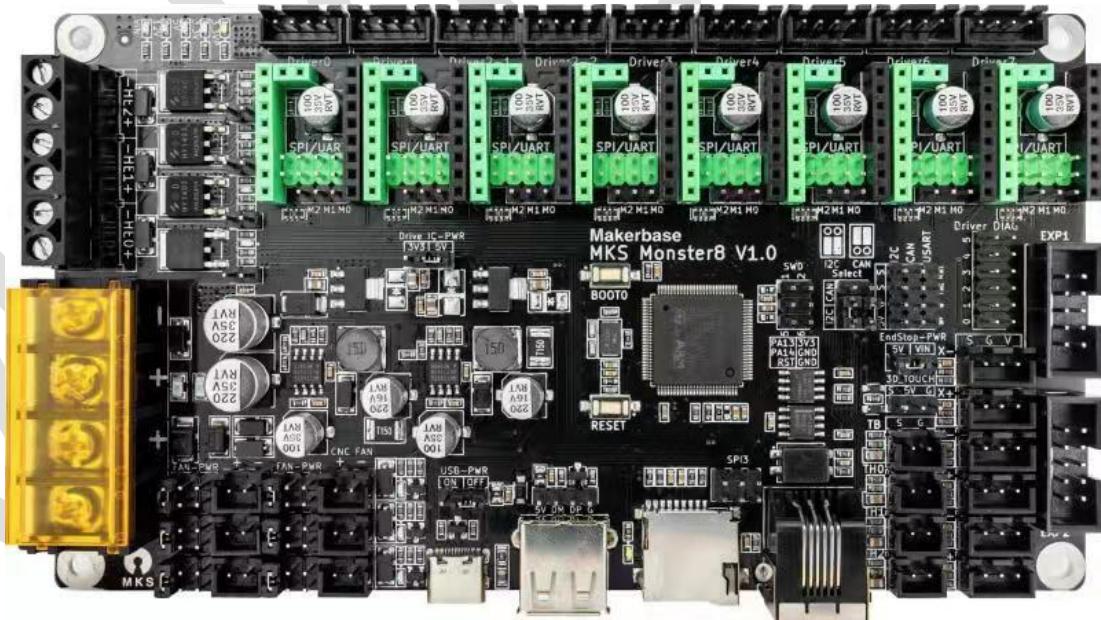




Guangzhou Qianhui Information Technology Co., Ltd.

MKS MONSTER8 V1.0 datasheet

*((based on Marlin firmware to configure
Voron 2.4 machine))*



About us:

Facebook: MAKERBASE

(Welcome to join our group to discuss issues together)

YouTub: Makerbase Team

(Welcome to subscribe to our account, we will continue to update the company's product video tutorials)

Github: makerbase-mks

AliExpress:

<https://www.aliexpress.com/store/1047297>

Amazon:

https://www.amazon.com/s?me=A25AM6LC3BZ7LE&fbclid=IwAR1q7Z7g0w6nS0xWC6Z6eyVqgR9hCTN_EF3YoYbcrIG5kX_gZ7KfDR-9fg&marketplaceID=ATVPDKIKX0DER

contents

About us:.....	2
Facebook:	2

YouTub:.....	2
Github:.....	2
AliExpress:	2
Amazon:.....	2
1.Product Brief.....	7
1.1 Features and advantages.....	7
1.2 Motherboard parameters.....	8
1.3 Wiring diagram.....	10
1.3.1 Wiring diagram of each port on the motherboard.....	10
1.3.2 Motor wiring based on Voron 2.4.....	11
1.3.3 Wiring of the leveling sensor PL08N on the board.....	13
1.3.4 Wiring diagram based on Voron2.4 3Dtouch.....	13
1.3.5 Limit wiring based on Voron 2.4.....	15
1.4 Dimensions.....	15
2. Firmware download, compile and update.....	16
2.1 Firmware download.....	16
2.2 Firmware compilation.....	16
2.2.1 Platformio local compilation.....	16
2.2.2 Web page online compilation.....	16
2.3 Firmware update.....	17
3.drive jumper setting.....	18
3.1 A4988 driver jumper setting.....	19

3.2 TMC2208, TMC2209, TMC2226 common mode jumper	19
3.3 TMC2225 common mode jumper setting.....	20
3.4 TMC2208, TMC2209, TMC2225, TMC2226 UART mode jumper settings.....	21
4. uart mode drive current setting.....	22
4.1 Set the current on the LCD screen.....	23
4.2 Set the current in the advanced configuration file.....	23
5. Marlin firmware configuration.....	24
5.1 Basic firmware configuration (must configure items).....	24
Note: Use the default firmware of mks monster8, download the mks monter8 firmware from the mks monster8 github page , these configuration items are already configured, do not need to be configured, if the firmware downloaded from marlin github needs to be configured.....	24
5.1.1 Board configuration.....	24
5.1.2 Serial port configuration.....	25
5.1.3 LCD configuration.....	25
5.1.4 Enable SD card.....	26
5.1.5Enable eeprom.....	27
5.1.6 Configure Compilation Environment.....	28
5.2 Machine parameter configuration (set according to machine parameter).....	28

5.2.1 Number of extrusion heads configuration.....	28
5.2.2 Thermal type configuration.....	29
5.2.3 Limit switch configuration.....	30
5.2.4 Pulse setting.....	31
5.2.5 Machine type configuration.....	32
5.2.6 Zero return direction setting.....	33
5.2.7 Print platform range setting.....	33
5.2.8 4Z motor configuration.....	34
5.2.9 Motor direction setting.....	35
6. Advanced configuration.....	39
6.1 Resume after power off (default firmware has been enabled)	39
6.2 Filament of runout sensor.....	40
6.3.3 TMC 2208/TMC 2209/TMC 2225/TMC 2226 uart mode	44
7. 4Z automatic leveling (sensor is PL08N).....	45
7.1 Set the sensor trigger level.....	45
7.2 Set sensor signal pin.....	46
7.3 Offset setting.....	48
7.4 Configure the probe type.....	48
7.5 Enable 4Z automatic leveling.....	49
7.6 Set the probe position of 4 motors.....	50
7.7 Set the number of leveling.....	51

7.8 z_safe_homing configuration.....	51
8.4Z automatic leveling (sensor is 3Dtouch).....	53
8.1 Set the sensor trigger level.....	53
8.2 Set sensor signal pin.....	54
8.3 Configure the probe type (enable BLTOUCH).....	55
8.4 Set the offset between the probe and the extrusion head.....	56
8.5 Set the distance between the sensor and the edge of the printing platform.....	57
8.6 Enable the servos.....	58
8.7 Set servo pin port.....	58
8.8 Enable 4Z automatic leveling.....	59
8.9 Set the detection position of 4 motors.....	60
8.10 Set the number of leveling.....	61
8.11 z_safe_homing configuration.....	61
9. automatic leveling operation on LCD screen.....	63
10. z_offset adjustment.....	64
10.1 z_offset adjustment wizard configuration.....	64
10.2 Operate on the LCD screen to adjust z_offset.....	64
11. LCD shell, motherboard mounting bracket stl file download link....	65
12. FAQ.....	65

1. Product Brief

MKS MONSTER8 V1.0 motherboard is a motherboard launched by the makerbase team to meet market needs. It can be used on Voron 2.4 machines, supports Marlin firmware and Klipper firmware, supports U disk printing (for the time being only supported by Marlin firmware), and supports TMC driver UART mode .

1.1 Features and advantages

1. TVS power spike processing to better protect the back-end circuit and load;
2. 3 channels of controllable fan output, the output voltage is adjustable, respectively adjustable to 5V, 12V, 24V;
3. Support U disk printing (currently only supported by Marlin firmware);
4. The user can replace the motor drive by himself, supporting A4988, DRV8825, LV8729, TMC2208, TMC2209, TMC2225, TMC2226;
5. Using high-quality MOSFET tubes, the heat dissipation effect is better, and the long-term work is stable;
6. Adopt dedicated power chip, support 12V-24V power input,
7. The stable and reliable filter circuit greatly reduces the possibility of

interference, and avoids crashes and random running during the printing process to the greatest possible extent;

8. Use open source firmware Marlin and Klipper;

9. Support LCD2004, LCD12864, MKS MINI12864 V1.0, MKS MINI12864 V3.0, support MKS series touch screen and H43 touch screen developed by maker;

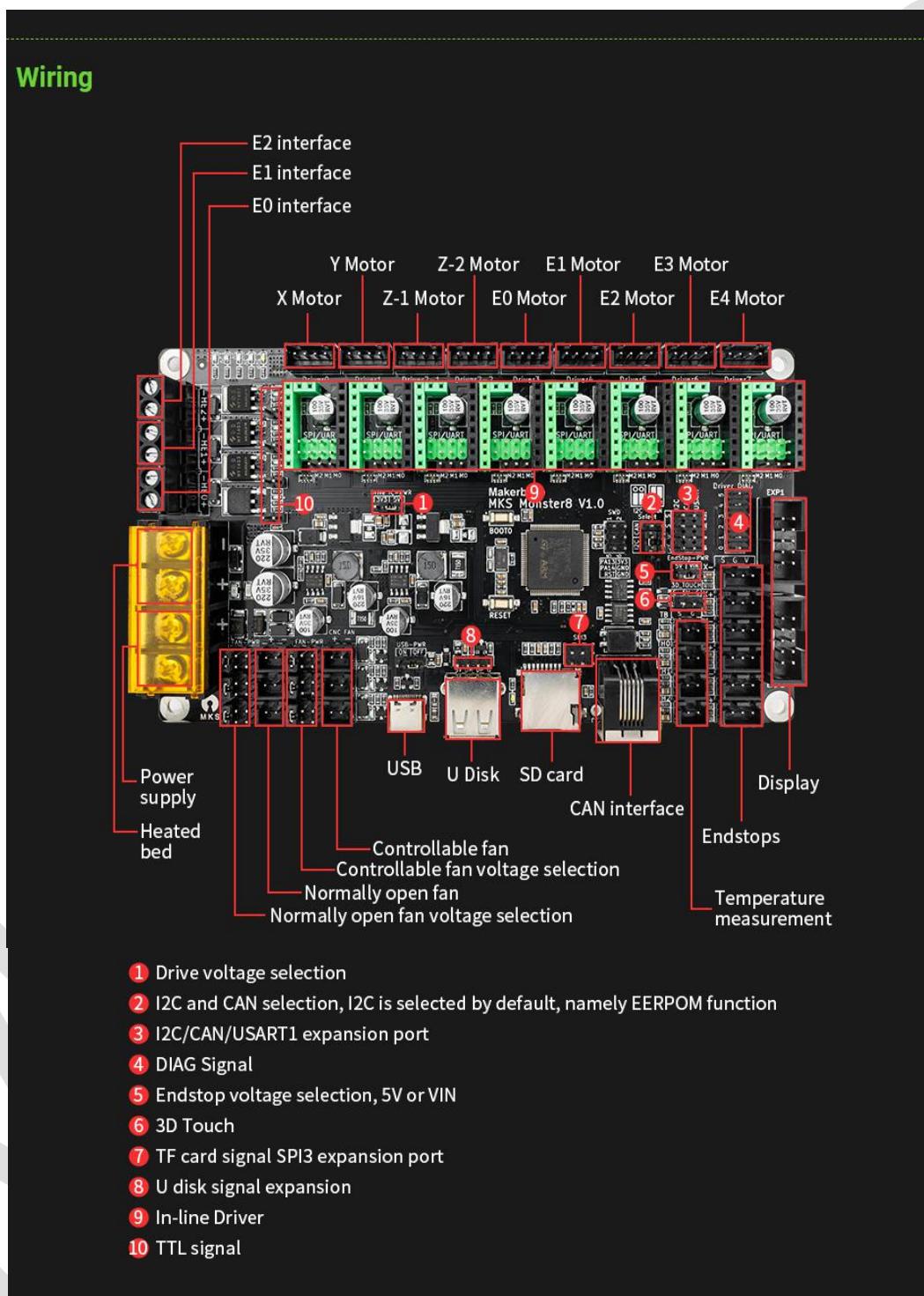
10. Support TMC2130 drive SPI mode, TMC2208, 2209, TMC2225, 2226 UART mode, support TMC2209, TMC222 unlimited bit reset..

1.2 Motherboard parameters

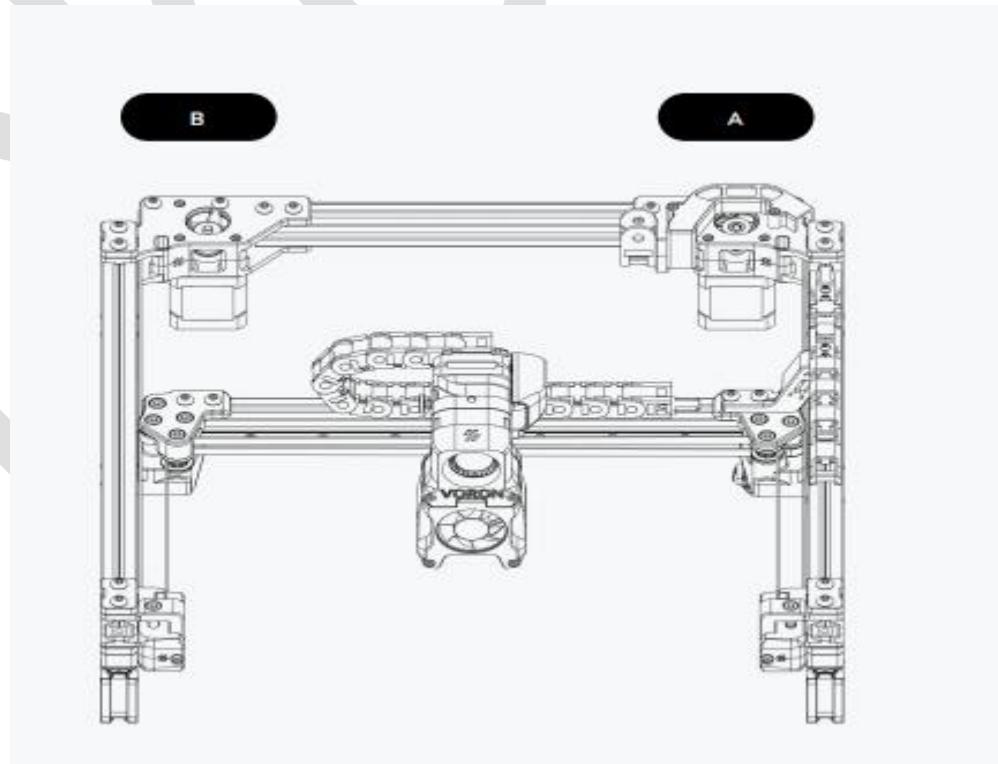
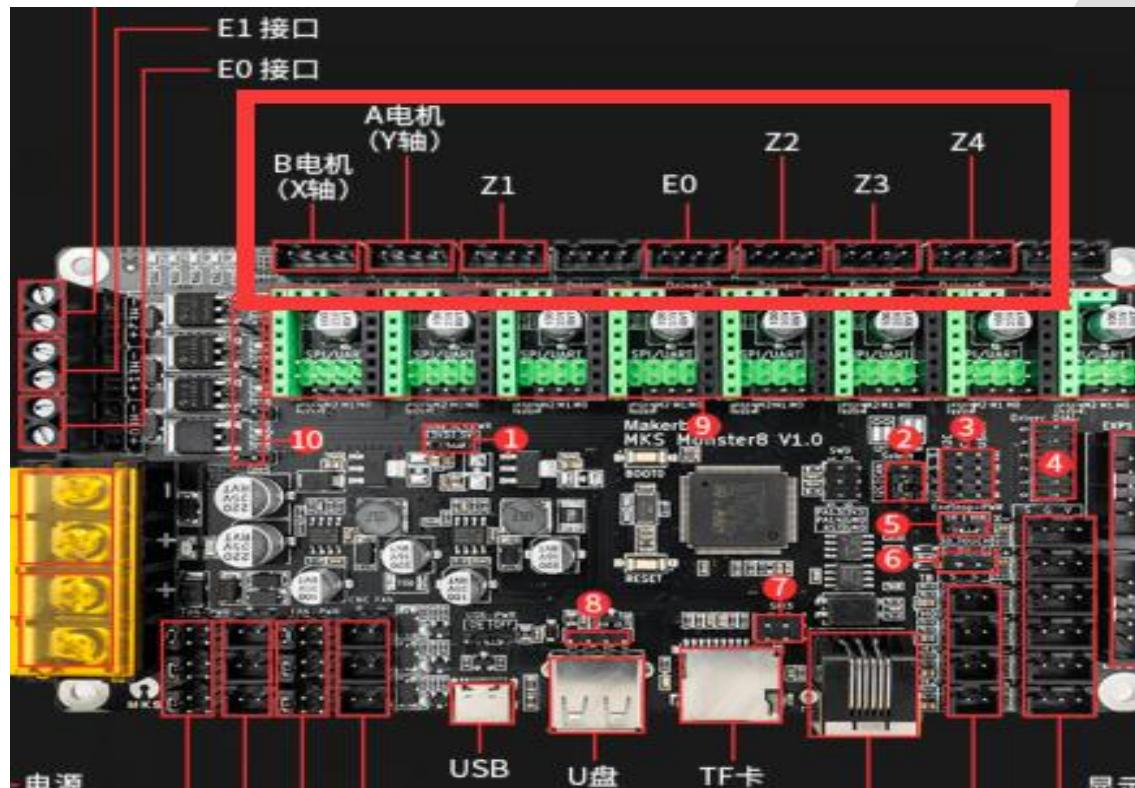
Motherboard	MKS MONSTER8	MCU:	STM32F407VET6
model:	V1.0		
physical dimension:	160mm*90mm	Mounting hole size:	152mm*82mm
Input voltage:	12V~24V 5A~20A	motor driver:	TMC2208,TMC2209,TMC2225,TMC2226,A4988,DRV8825,LV8729
Temperature sensor interface:	NTC 100K	Support LCD/touch screen	LCD2004、LCD12864、MKS MINI12864 V1.0、MKS MINI12864 V3.0、MKS TFT Series touch screen
Support print file format:	G-code	Support machine structure:	XYZ、delta、kossel、Ultimaker、corexy
Recommended software:	Cura、Simplify3d、Pronterface、Repetier-Host	Firmware update:	TF card

1.3 Wiring diagram

1.3.1 Wiring diagram of each port on the motherboard



1.3.2 Motor wiring based on Voron 2.4





Note: Please be sure to follow the above wiring. Wrong wiring of A and

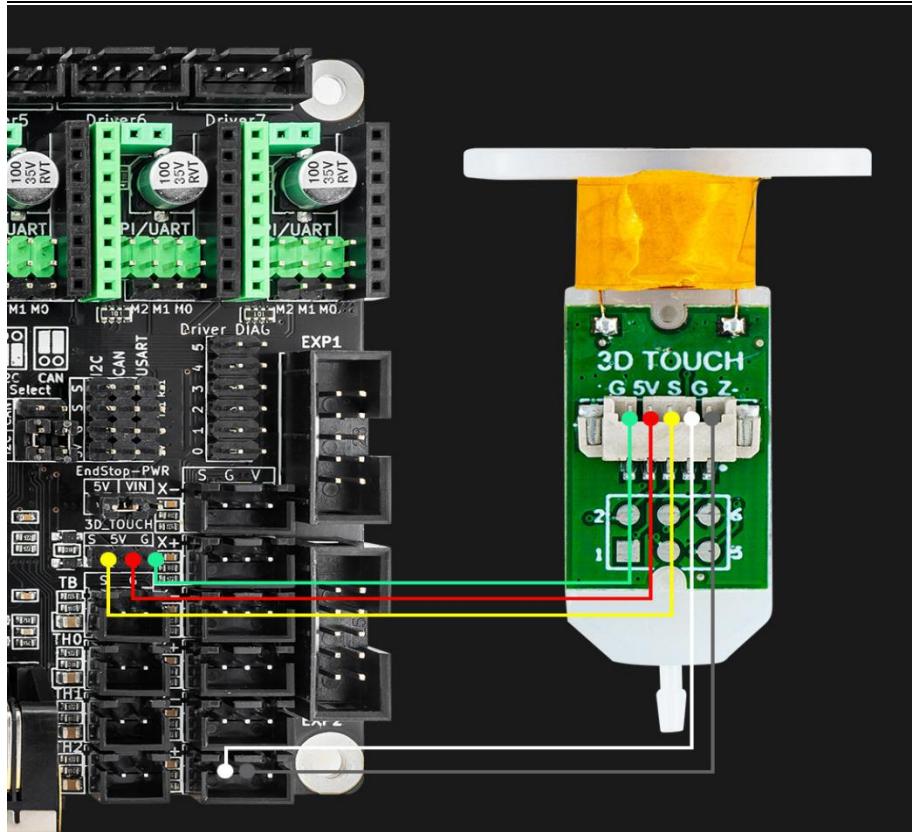
B motors will cause movement errors, printing mirror images, etc., wrong 4Z sequence connection will cause incorrect leveling.

1.3.3 Wiring of the leveling sensor PL08N on the board

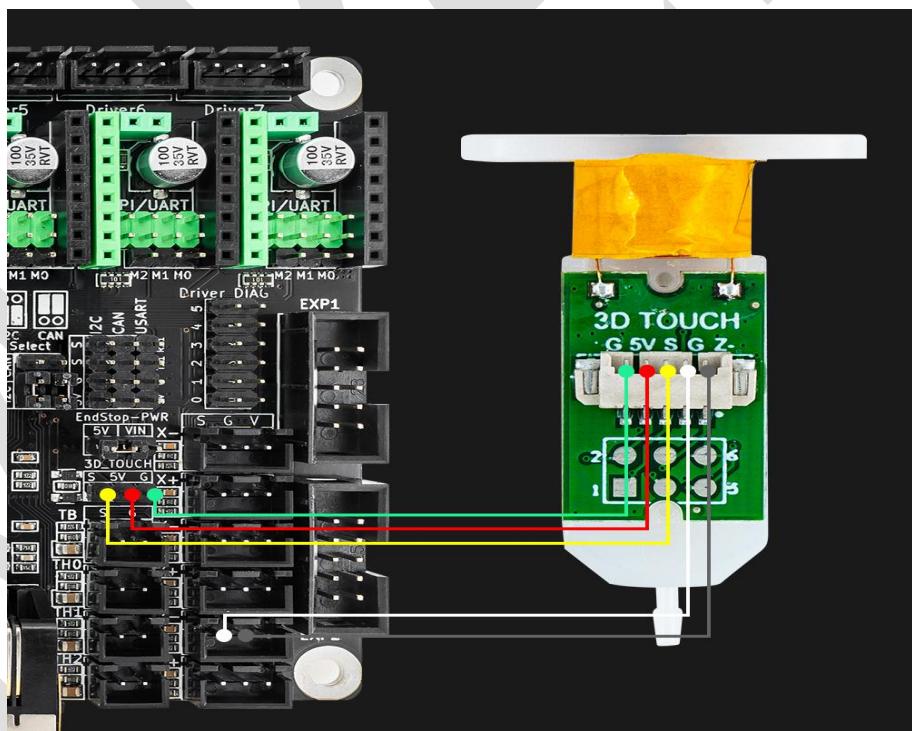
The three wires of PL08N are brown to the positive pole of the power supply, blue to the negative pole of the power supply, and black to the leveling interface signal pin; when PL08N is only used for leveling, the signal line (black) of PL08N is connected to the Z_MAX limit S terminal; when PL08N Used as zero return Z limit (when z_safe_homing function, the signal line (black) of PL08N is connected to Z_MIN limit.

1.3.4 Wiring diagram based on Voron2.4 3Dtouch

When 3Dtouch is only used for leveling, the signal line of 3D touch is connected to the Z_MAX signal pin



When 3Dtough is used as a limit (`z_safe_homing`), the signal line of 3D touch is connected to the `Z_MIN` signal pin

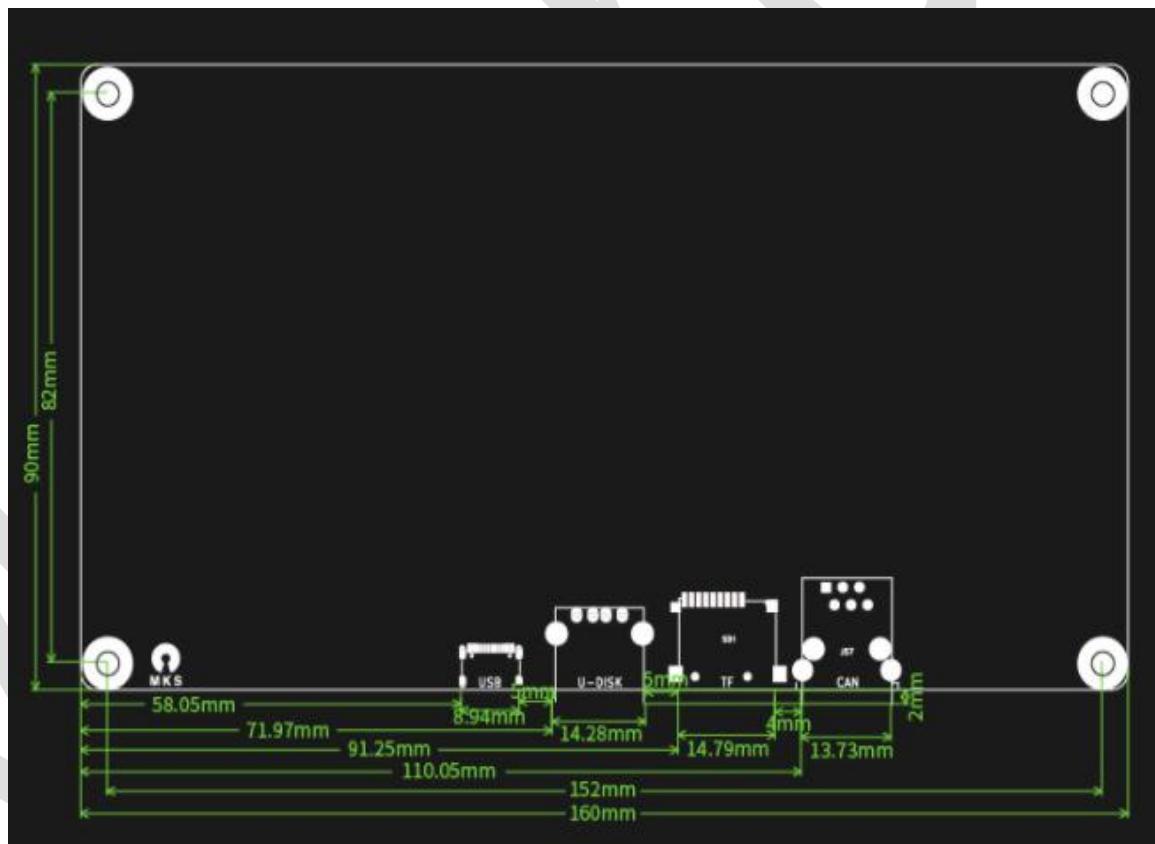


1.3.5 Limit wiring based on Voron 2.4

Based on Voron 2.4, the home position of X axis and Y axis is the upper right corner, that is, the home direction of X axis and Y axis is to the maximum direction, then X axis and Y axis limit are connected to X_MAX and Y_MAX limit.

1.4 Dimensions

Motherboard size chart:



2. Firmware download, compile and update

2.1 Firmware download

The firmware that MKS MONSTER8 V1.0 adapts to is Marlin2.X firmware. The firmware can be downloaded from the official Github of MAKERBASE. The firmware download address:

<https://github.com/makerbase-mks/MKS-Monster8/tree/main/marlin%20firmware%20for%20voron>

2.2 Firmware compilation

2.2.1 Platformio local compilation

Platformio local compilation needs to install VScode software and install platformio compilation plug-in first

2.2.2 Web page online compilation

Online compilation, you can log in to the Baizhongyun webpage of the maker base, upload the compressed package of the firmware, and compile online. After the successful compilation, the firmware file will be sent to your mailbox, which is convenient and fast. Compile the website online: <https://baizhongyun.cn/home/index>

2.3 Firmware update

Compiled by local VScode, the .bin file after successful compilation is in the .pio folder:

D (D:) > zuixin marlin gujian > MKS Monster8 marlin > Marlin-2.0.x > .pio > build > mks_monster8_usb_flash_drive_msc >			
名称	修改日期	类型	大小
FrameworkArduino	2021/9/2 15:10	文件夹	
FrameworkArduinoVariant	2021/9/2 15:09	文件夹	
lib5df	2021/9/2 15:13	文件夹	
lib23a	2021/9/2 15:13	文件夹	
lib44a	2021/9/2 15:13	文件夹	
lib101	2021/9/2 15:13	文件夹	
lib228	2021/9/2 15:13	文件夹	
lib289	2021/9/2 15:13	文件夹	
lib482	2021/9/2 15:13	文件夹	
libbcd	2021/9/2 15:13	文件夹	
libff8	2021/9/2 15:13	文件夹	
src	2021/9/2 15:11	文件夹	
SrcWrapper	2021/9/2 15:10	文件夹	
.gcc_path	2021/9/2 15:06	GCC_PATH 文件	1 KB
.sconsign37.dblite	2021/9/7 16:57	DBLITE 文件	14,337 KB
firmware.bin	2021/9/7 16:57	BIN 文件	202 KB
firmware.elf	2021/9/7 16:57	ELF 文件	21,454 KB
longcmd-32093138880daa52c11b06...	2021/9/2 15:13	文件	13 KB
mks_monster8.bin	2021/9/7 16:57	BIN 文件	202 KB

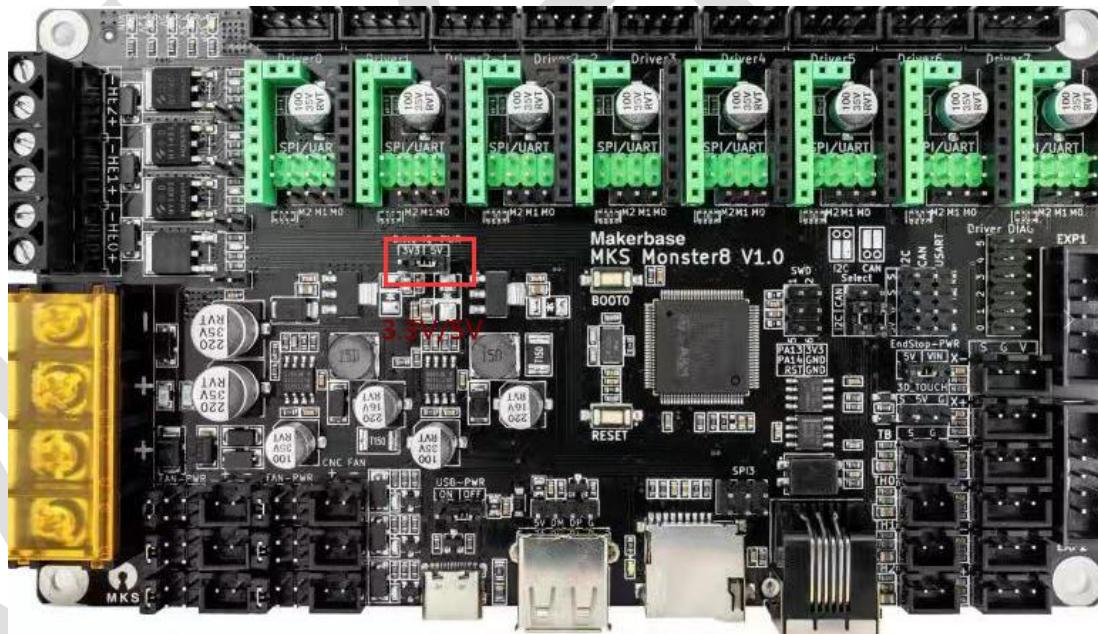
Compiled on the webpage, the mks_monster8.bin file after the compilation is successful will be sent to the mailbox filled in on the webpage;

Copy the mks_monster8.bin file to the TF card, insert the card into the TF card slot of the motherboard, then power on and wait for the firmware update; after the update is successful, the mks_monster8.bin in the TF card will be renamed to MKS_MONSTER8.CUR , If you need to update the firmware again, you need to delete the MKS_MONSTER8.CUR file

in the TF card first, and then copy the mks/_monster8.bin file to the TF card.

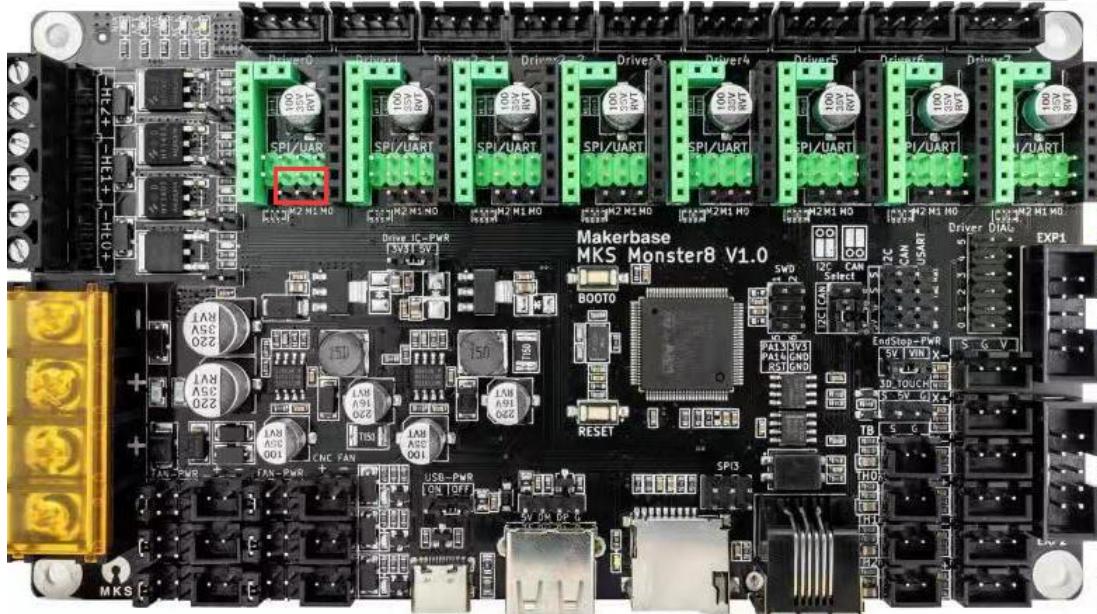
3. drive jumper setting

Note: The voltage supplied to the driver on the motherboard can be set by jumpers. It can be set to 3.3 or 5V. It must be set to one of them. If the driver is not set, it will not work. It is recommended to set to 5V. (The sensorless function of the old version of marlin firmware is required Set to 3.3V) When set to 3.3V, the voltage driven by the A4988 will be halved and the current will also be halved.



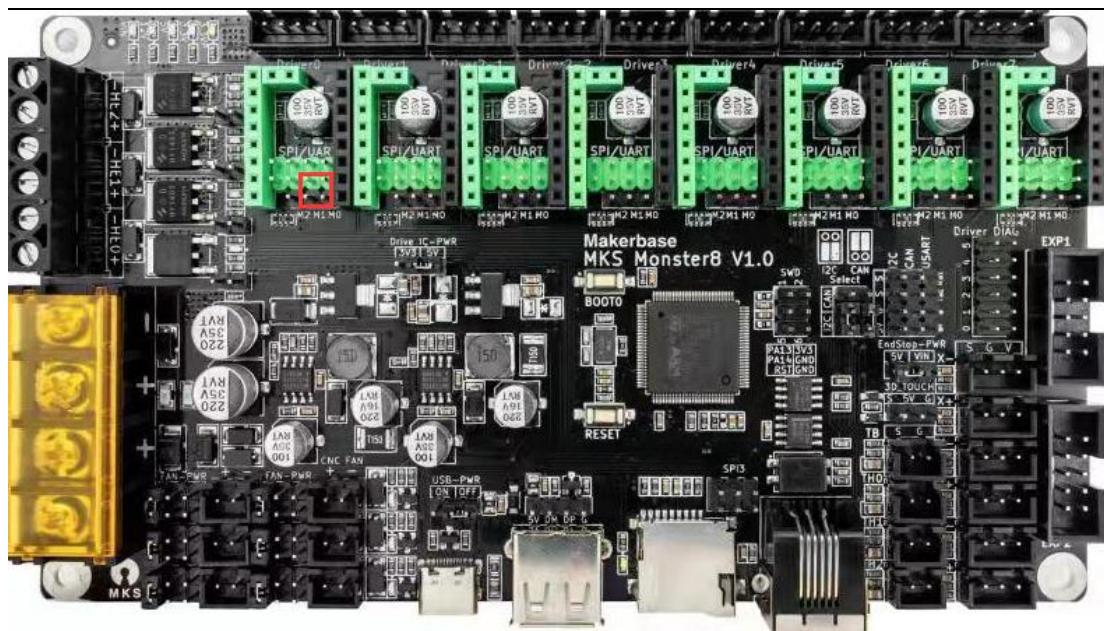
3.1 A4988 driver jumper setting

A4988 drives the subdivision jumper mode, the 3 jumper caps below the driver are plugged into 16 subdivisions, as shown in the following figure (X-axis as an example):



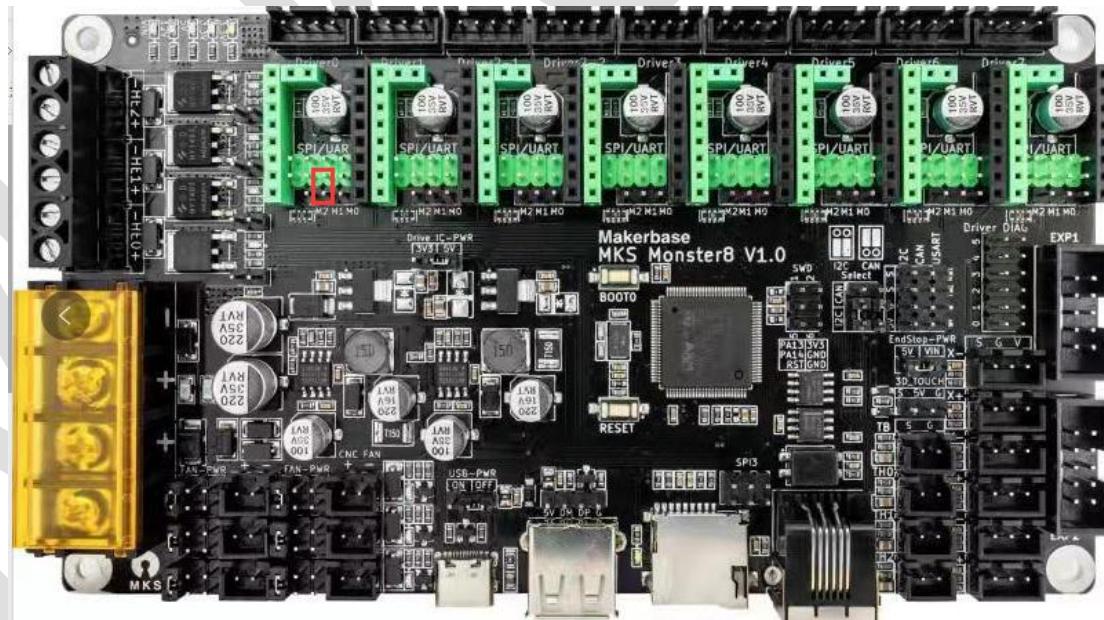
3.2 TMC2208, TMC2209, TMC2226 common mode jumper

Settings TMC2208, TMC2209, TMC2226 drive the subdivision jumper mode, the 2 jumper caps (M0, M1) below the driver are plugged into 16 subdivisions, as shown in the following figure (X-axis as an example):



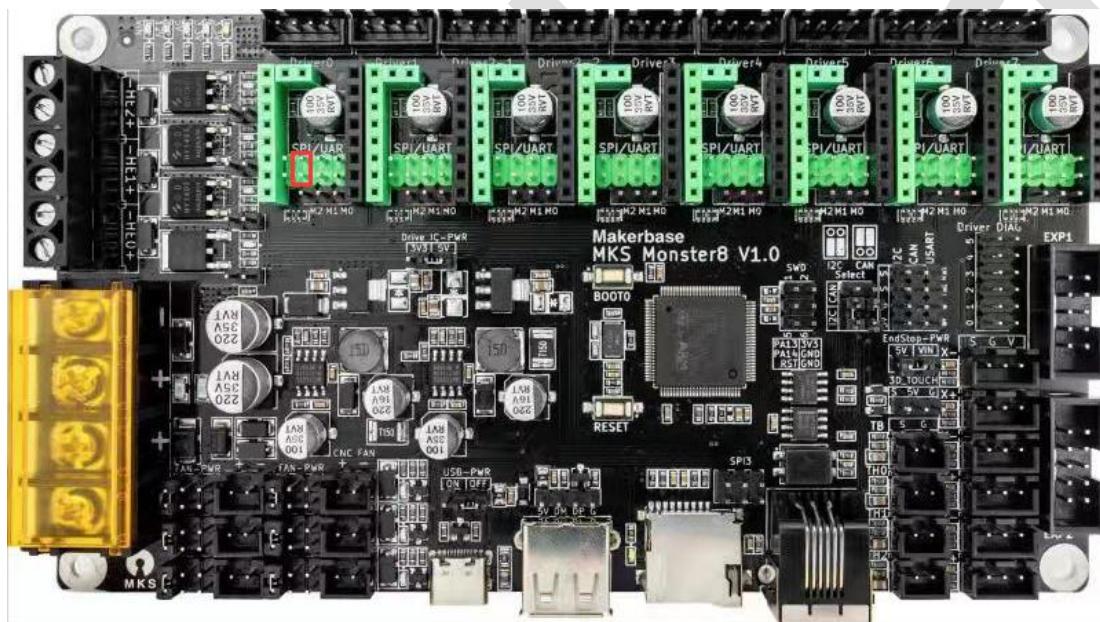
3.3 TMC2225 common mode jumper setting

TMC2225 drives the subdivision jumper mode, and the second jumper cap (M1) on the right under the drive is inserted into 16 subdivisions, as shown in the following figure (X-axis as an example):

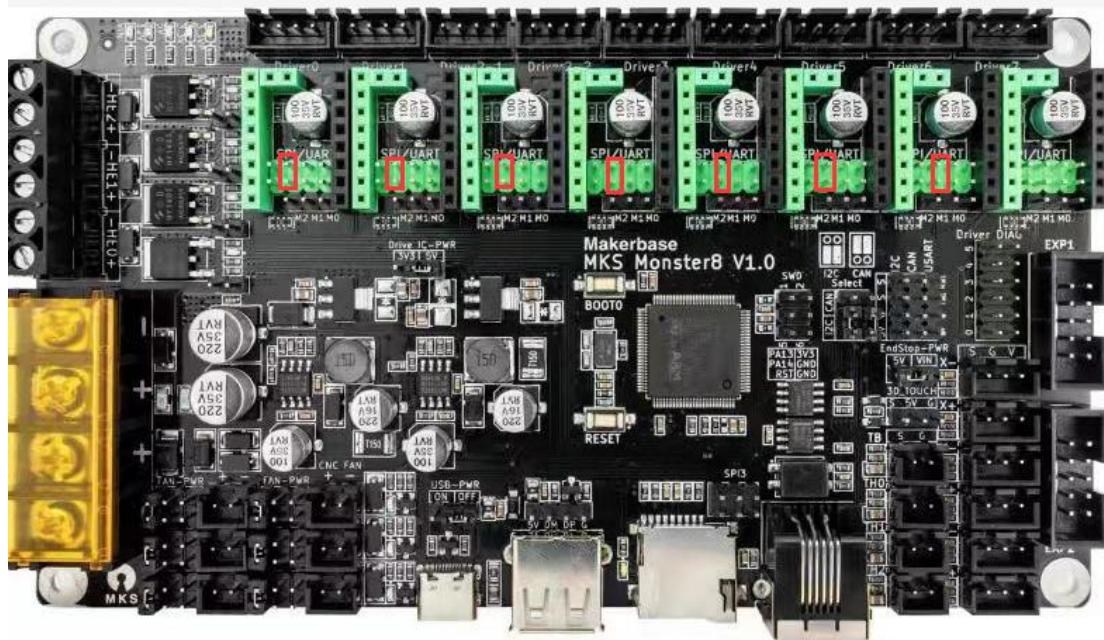


3.4 TMC2208, TMC2209, TMC2225, TMC2226 UART mode jumper settings

TMC2208, TMC2209, TMC2225, TMC2226 UART mode jumper settings are the same, the third jumper on the left under the driver is plugged into uart mode, as shown in the figure below (X-axis as an example):



Note: Based on Voron V2.4, in order to facilitate the drive current adjustment, the default firmware of the mks monter8 V1.0 motherboard (firmware obtained from the mks qq group or mks github) is to enable 7 TMC2209 UART modes, so use it directly. The default firmware requires 7 drivers to be set to uart mode.



Note: The use of UART mode also needs to enable the TMC driver in the firmware. Please refer to 6.3.3 for the configuration method.

4. uart mode drive current setting

The current setting needs to be configured according to the rated current of the motor you use. The rated current (peak current) of the motor divided by 1.41 is the current set on the LCD screen or advanced configuration file. Some motors have larger coil internal resistance. Configuring according to the rated current of the motor may cause serious heating of the motor. At this time, you need to reduce the current. It is recommended to set the current to 80% to 90% of the rated current of the motor, for example:

The rated current of the motor is 1.2A, and the current can be set to 1A,

then the current set on the LCD or in the advanced configuration file is:

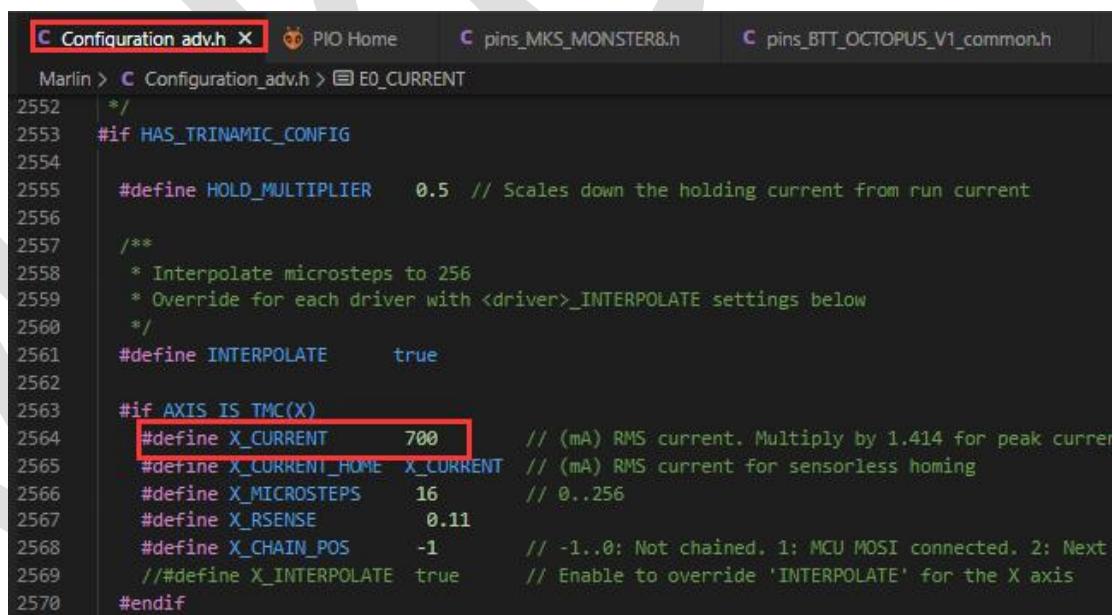
$$1A/1.41=0.7A$$

4.1 Set the current on the LCD screen

Press the rotary encoder → configuration → advanced configuration → TMC drive → drive current on the screen to configure the current. After the configuration, you need to return to the configuration interface to save and load the parameters:

4.2 Set the current in the advanced configuration file

Find the TMC drive current configuration item in the advanced configuration file, and then modify the current, take the X axis as an example, set 700mA.



```
Configuration_adv.h X PIO Home pins_MKS_MONSTER8.h pins_BITT_OCTOPUS_V1_common.h
Marlin > Configuration_adv.h > EQ_CURRENT
2552 */
2553 #if HAS_TRINAMIC_CONFIG
2554
2555 #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2556
2557 /**
2558 * Interpolate microsteps to 256
2559 * Override for each driver with <driver>_INTERPOLATE settings below
2560 */
2561 #define INTERPOLATE      true
2562
2563 #if AXIS_IS_TMC(X)
2564 #define X_CURRENT        700 // (mA) RMS current. Multiply by 1.414 for peak current
2565 #define X_CURRENT_HOME   X_CURRENT // (mA) RMS current for sensorless homing
2566 #define X_MICROSTEPS     16 // 0..256
2567 #define X_RSENSE          0.11
2568 #define X_CHAIN_POS       -1 // -1..0: Not chained. 1: MCU MOSI connected. 2: Next
2569 // #define X_INTERPOLATE   true // Enable to override 'INTERPOLATE' for the X axis
2570#endif
```

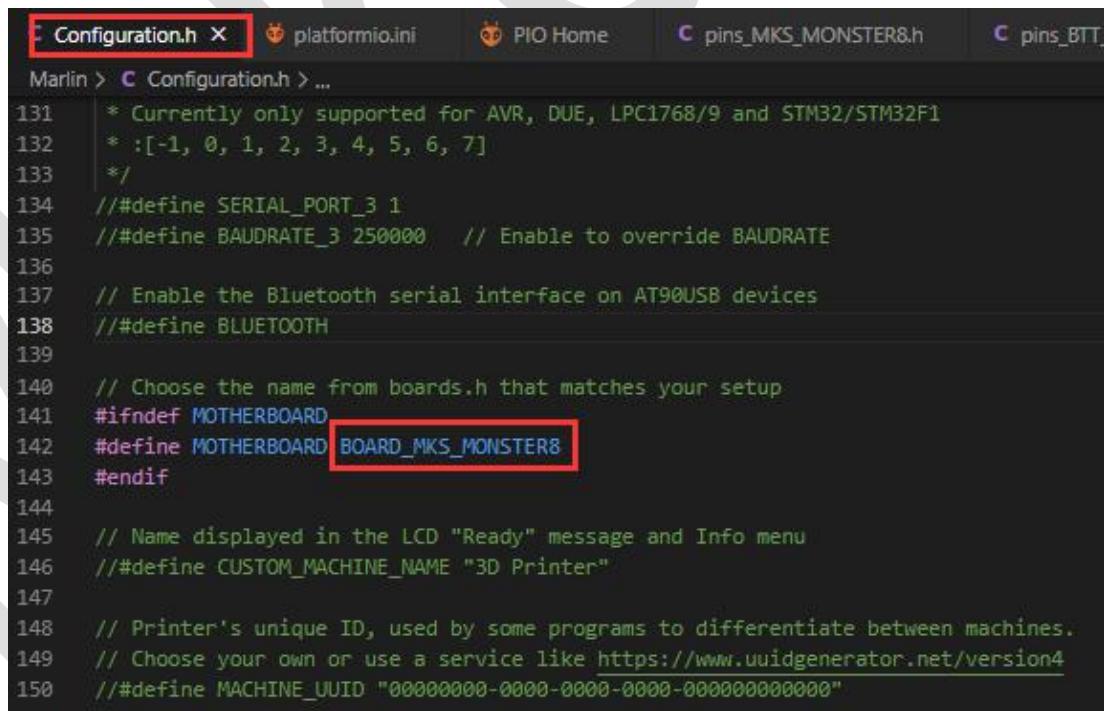
5. Marlin firmware configuration

5.1 Basic firmware configuration (must configure items)

Note: Use the default firmware of mks monster8, download the mks monter8 firmware from the mks monster8 github page , these configuration items are already configured, do not need to be configured, if the firmware downloaded from marlin github needs to be configured

5.1.1 Board configuration

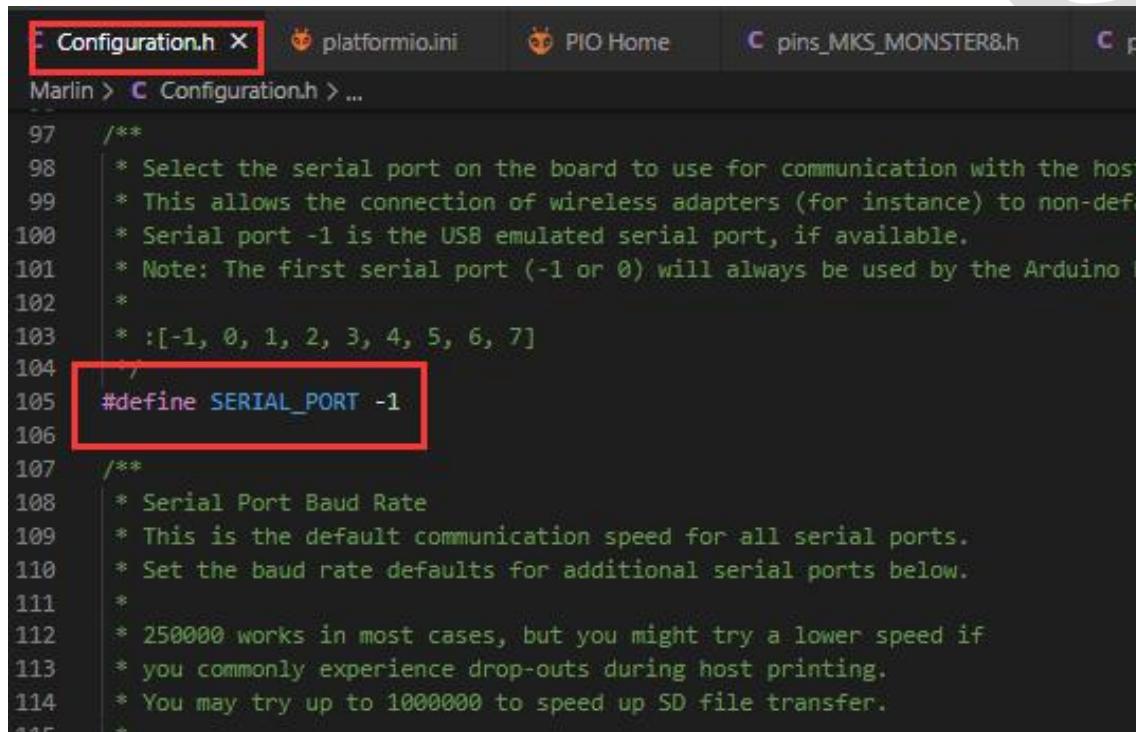
Configure the board as BOARD_MKS_MONSTER8 in the configuration.h file



```
Configuration.h X platformio.ini PIO Home pins_MKS_MONSTER8.h pins_BTT.h
Marlin > C Configuration.h > ...
131 * Currently only supported for AVR, DUE, LPC1768/9 and STM32/STM32F1
132 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
133 */
134 //#define SERIAL_PORT_3 1
135 //#define BAUDRATE_3 250000 // Enable to override BAUDRATE
136
137 // Enable the Bluetooth serial interface on AT90USB devices
138 //#define BLUETOOTH
139
140 // Choose the name from boards.h that matches your setup
141 #ifndef MOTHERBOARD
142 #define MOTHERBOARD BOARD_MKS_MONSTER8
143 #endif
144
145 // Name displayed in the LCD "Ready" message and Info menu
146 //#define CUSTOM_MACHINE_NAME "3D Printer"
147
148 // Printer's unique ID, used by some programs to differentiate between machines.
149 // Choose your own or use a service like https://www.uuidgenerator.net/version4
150 //#define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
```

5.1.2 Serial port configuration

Set the serial port to -1 in the configuration.h file, and configuration errors will cause the motherboard to fail to connect with the host computer.



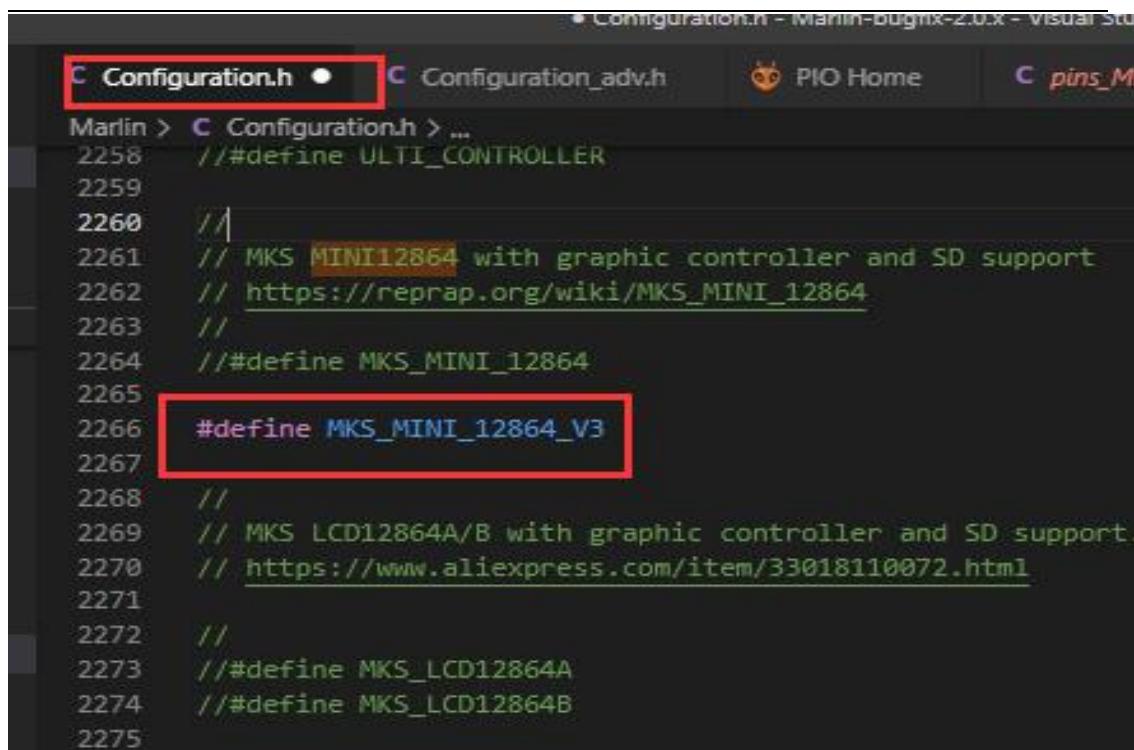
```
Configuration.h X platformio.ini PIO Home pins_MKS_MONSTER8.h pins_MKS_VORON8.h
Marlin > Configuration.h > ...
97 /**
98  * Select the serial port on the board to use for communication with the host
99  * This allows the connection of wireless adapters (for instance) to non-default
100 * Serial port -1 is the USB emulated serial port, if available.
101 * Note: The first serial port (-1 or 0) will always be used by the Arduino
102 *
103 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
104 */
105 #define SERIAL_PORT -1
106 /**
107  * Serial Port Baud Rate
108  * This is the default communication speed for all serial ports.
109  * Set the baud rate defaults for additional serial ports below.
110 *
111 *
112 * 250000 works in most cases, but you might try a lower speed if
113 * you commonly experience drop-outs during host printing.
114 * You may try up to 1000000 to speed up SD file transfer.
115 *
```

Note: The connection between the upper computer and the main board of the computer requires that the baud rate set by the upper computer is consistent with that set by the main board, and the default baud rate is 250,000.

5.1.3 LCD configuration

Based on VORON 2.4, the LCD screen matched with mks monster8 V1.0 motherboard is mks mini12864 V3.0

Enable MKS in the configuration file _MINI12864_V3



```
* Configuration.h - Marlin-2019x-2.0.0 - Visual Studio Code
C Configuration.h • C Configuration_adv.h    PIO Home    C pins_M...
Marlin > C Configuration.h > ...
2258 // #define ULTI_CONTROLLER
2259
2260 //|
2261 // MKS MINI12864 with graphic controller and SD support
2262 // https://reprap.org/wiki/MKS\_MINI\_12864
2263 //
2264 // #define MKS_MINI_12864
2265
2266 #define MKS_MINI_12864_V3
2267
2268 //|
2269 // MKS LCD12864A/B with graphic controller and SD support.
2270 // https://www.aliexpress.com/item/33018110072.html
2271 //
2272 //|
2273 // #define MKS_LCD12864A
2274 // #define MKS_LCD12864B
2275
```

Note: The size of MKS MINI12864 V3.0 LCD is inconsistent with that of Voron 2.4, so you can't print the LCD shell directly with Voron 2.4 stl file. Download link of MKS MINI12864 V3.0 shell:

<https://www.thingiverse.com/thing:4918948>

5.1.4 Enable SD card

Enable #define SDSUPPORT in the configuration file

```
C Configuration.h •
Marlin > C Configuration.h > SDSUPPORT
2062 /**
2063 * SD CARD
2064 *
2065 * SD Card support is disabled by default. If your controller
2066 * you must uncomment the following option or it won't work.
2067 */
2068 #define SDSUPPORT
2069 /**
2070 * SD CARD: ENABLE CRC
2071 *
2072 * Use CRC checks and retries on the SD communication.
2073 */
2074 // #define SD_CHECK_AND_RETRY
2075 /**
2076 /**
2077 */


```

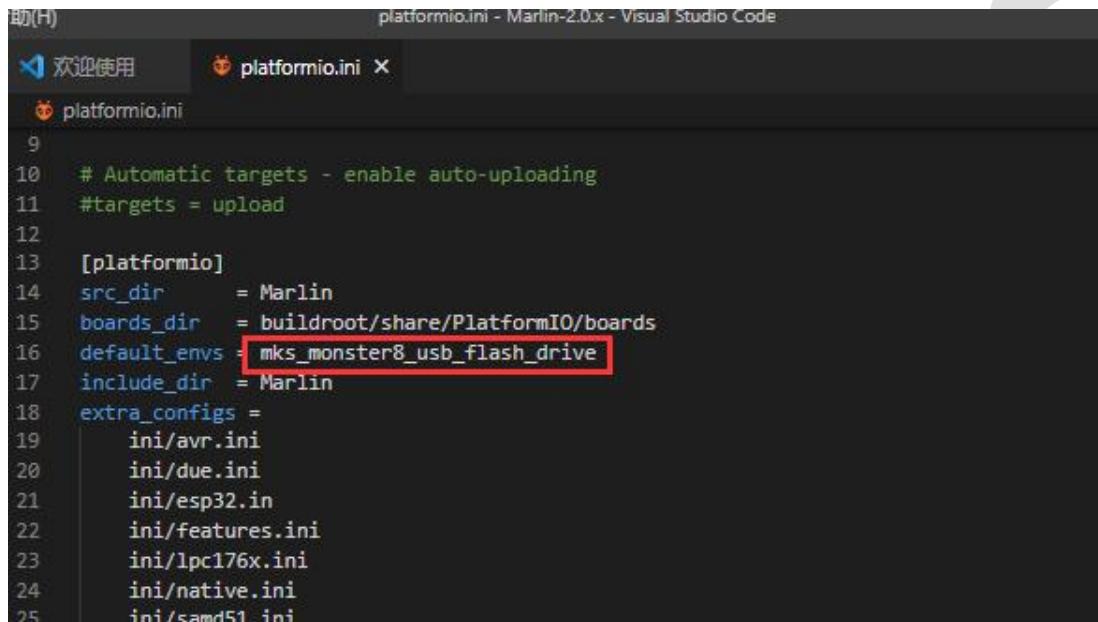
5.1.5 Enable eeprom

Enable `#define EEPROM_SETTINGS` in the configuration file

```
C Configuration.h •
Marlin > C Configuration.h > EEPROM_SETTINGS
1791 * M500 - Store settings to EEPROM.
1792 * M501 - Read settings from EEPROM. (i.e., Throw away un
1793 * M502 - Revert settings to "factory" defaults. (Follow w
1794 */
1795 #define EEPROM_SETTINGS // Persistent storage with M500
1796 // #define DISABLE_M503 // Saves ~2700 bytes of PROGME
1797 #define EEPROM_CHITCHAT // Give feedback on EEPROM com
1798 #define EEPROM_BOOT_SILENT // Keep M503 quiet and only gi
1799 #if ENABLED(EEPROM_SETTINGS)
1800 // #define EEPROM_AUTO_INIT // Init EEPROM automatically o
1801 #endif
1802 //
1803 //
1804 // Host Keepalive
1805 //
1806 // When enabled Marlin will send a busy status message to th
1807 // every couple of seconds when it can't accept commands.
```

5.1.6 Configure Compilation Environment

Change the compilation environment to mks_monster8_usb_flash_drive in the platformio.ini file



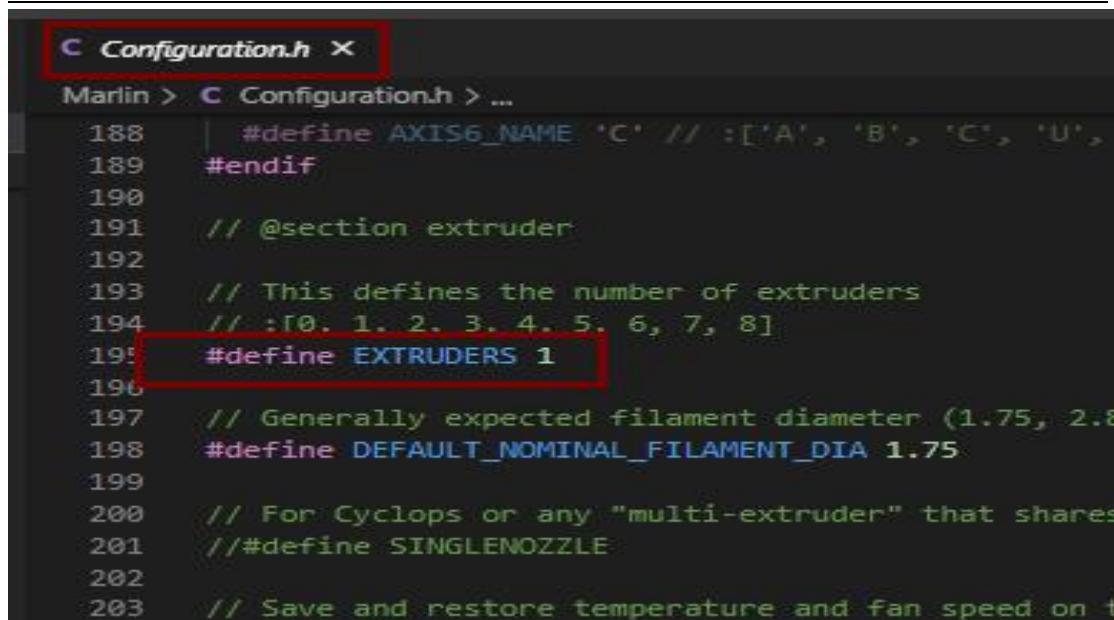
```
platformio.ini - Marlin-2.0.x - Visual Studio Code
platformio.ini

[platformio]
src_dir      = Marlin
boards_dir   = buildroot/share/PlatformIO/boards
default_envs = mks_monster8_usb_flash_drive
include_dir  = Marlin
extra_configs =
    ini/avr.ini
    ini/due.ini
    ini/esp32.in
    ini/features.ini
    ini/lpc176x.ini
    ini/native.ini
    ini/samd51.ini
```

5.2 Machine parameter configuration (set according to machine parameter)

5.2.1 Number of extrusion heads configuration

MKS MONSTER8 V1.0 supports up to 3 extrusion heads, based on Voron V2.4, just set to 1 extrusion head (the default firmware is set to 1)



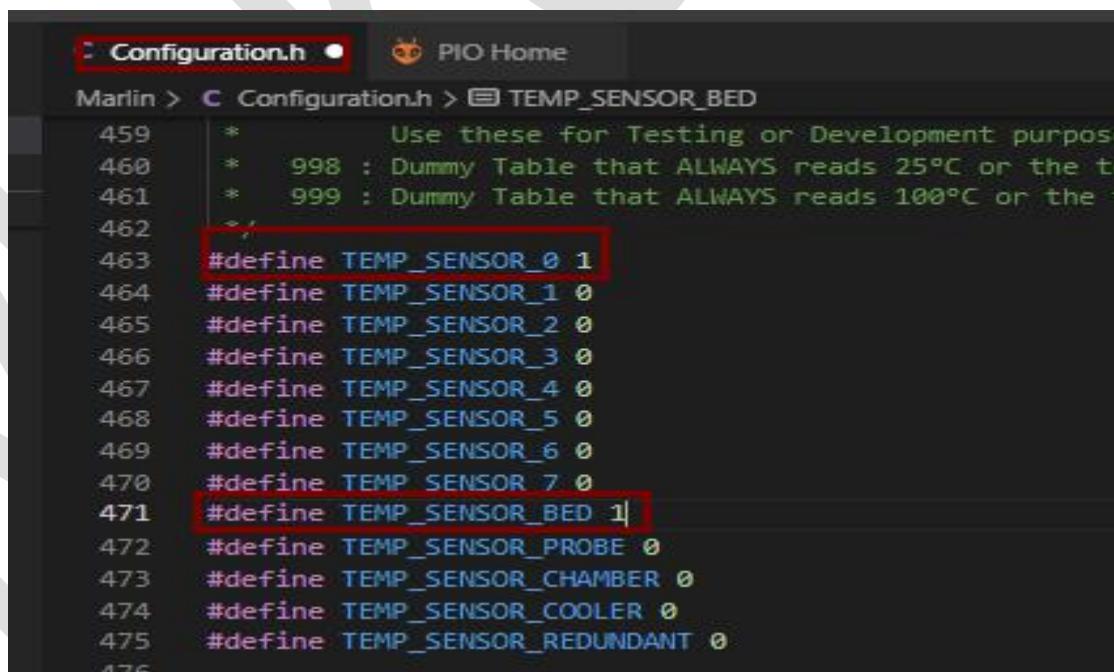
```

C Configuration.h ×
Marlin > C Configuration.h > ...
188     #define AXIS6_NAME 'C' // :['A', 'B', 'C', 'U',
189 #endif
190
191 // @section extruder
192
193 // This defines the number of extruders
194 // :[0, 1, 2, 3, 4, 5, 6, 7, 8]
195 #define EXTRUDERS 1
196
197 // Generally expected filament diameter (1.75, 2.8
198 #define DEFAULT_NOMINAL_FILAMENT_DIA 1.75
199
200 // For Cyclops or any "multi-extruder" that shares
201 // #define SINGLENOZZLE
202
203 // Save and restore temperature and fan speed on t

```

5.2.2 Thermal type configuration

MKS MONSTER8 only supports 100K thermal sensitivity, #define TEMP_SENSOR_0 is extrusion head 1, #define TEMP_SENSOR_BED is hot bed(Based on Voron2.4, the default firmware has enabled the hot bed and extruder 1 to be 100K thermal)



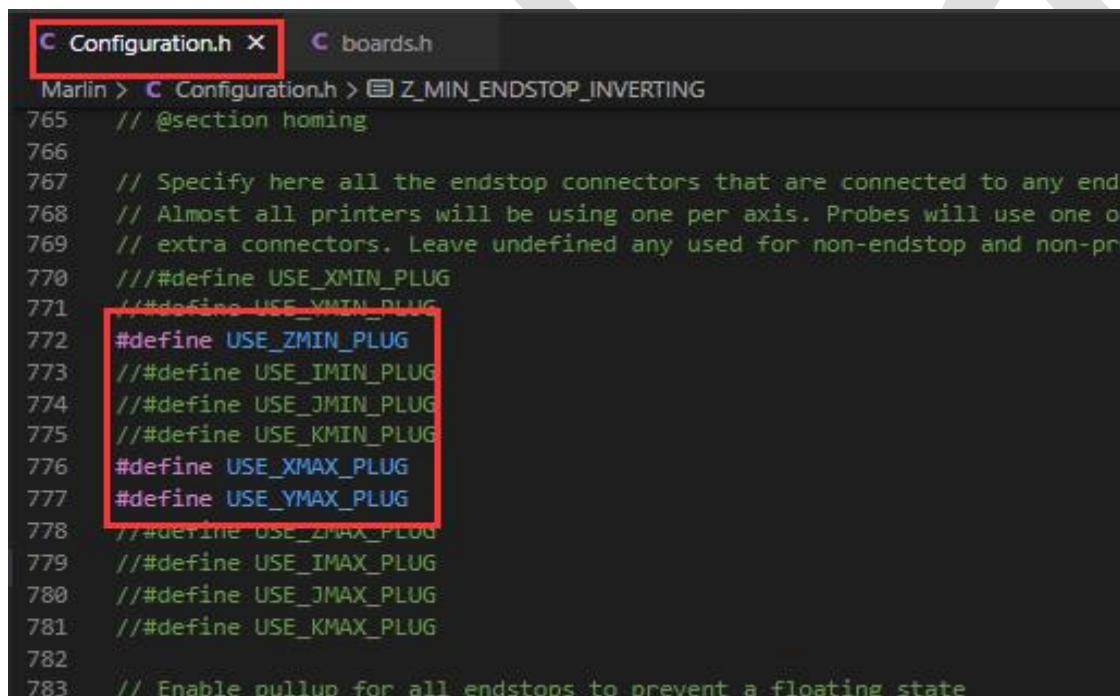
```

C Configuration.h • PIO Home
Marlin > C Configuration.h > TEMP_SENSOR_BED
459 *           Use these for Testing or Development purpose
460 *   998 : Dummy Table that ALWAYS reads 25°C or the te
461 *   999 : Dummy Table that ALWAYS reads 100°C or the t
462 */
463 #define TEMP_SENSOR_0 1
464 #define TEMP_SENSOR_1 0
465 #define TEMP_SENSOR_2 0
466 #define TEMP_SENSOR_3 0
467 #define TEMP_SENSOR_4 0
468 #define TEMP_SENSOR_5 0
469 #define TEMP_SENSOR_6 0
470 #define TEMP_SENSOR_7 0
471 #define TEMP_SENSOR_BED 1
472 #define TEMP_SENSOR_PROBE 0
473 #define TEMP_SENSOR_CHAMBER 0
474 #define TEMP_SENSOR_COOLER 0
475 #define TEMP_SENSOR_REDUNDANT 0
476

```

5.2.3 Limit switch configuration

Limit switch is enabled. Based on Voron 2.4, the home position of X axis and Y axis is the upper right corner, that is, the home direction of X axis and Y axis is to the maximum direction, then X axis and Y axis enable X_MAX and Y_MAX limit Bit; Z-axis return to the minimum direction, then enable the Z_MIN limit.



```
Configuration.h X boards.h
Marlin > Configuration.h > Z_MIN_ENDSTOP_INVERTING
765 // @section homing
766
767 // Specify here all the endstop connectors that are connected to any endstop
768 // Almost all printers will be using one per axis. Probes will use one or more
769 // extra connectors. Leave undefined any used for non-endstop and non-probe
770 // #define USE_XMIN_PLUG
771 // #define USE_YMIN_PLUG
772 #define USE_ZMIN_PLUG
773 // #define USE_IMIN_PLUG
774 // #define USE_JMIN_PLUG
775 // #define USE_KMIN_PLUG
776 #define USE_XMAX_PLUG
777 #define USE_YMAX_PLUG
778 // #define USE_ZMAX_PLUG
779 // #define USE_IMAX_PLUG
780 // #define USE_JMAX_PLUG
781 // #define USE_KMAX_PLUG
782
783 // Enable pullup for all endstops to prevent a floating state
```

Limit switch type configuration (true/false), true is normally open switch (NO), false is normally closed switch (NC), based on Voron 2.4, the limit is recommended to be set to normally closed switch (note: normally closed Type switch wiring mode is shown below)

```

C Configuration.h • C Configuration_adv.h      C pins_MKS_MONSTER8.h • C boards.h
Marlin > C Configuration.h > Z_MIN_PROBE_ENDSTOP_INVERTING
818 //#define ENDSTOPPULLDOWN_ZMIN_PROBE
819 #endif
820
821 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common)
822 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
823 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
824 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
825 #define I_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
826 #define J_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
827 #define K_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
828 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
829 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
830 #define Z_MAX_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.
831 #define I_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
832 #define J_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
833 #define K_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
834 #define Z_MIN_PROBE_ENDSTOP_INVERTING true // Set to true to invert the logic of the probe
835

```



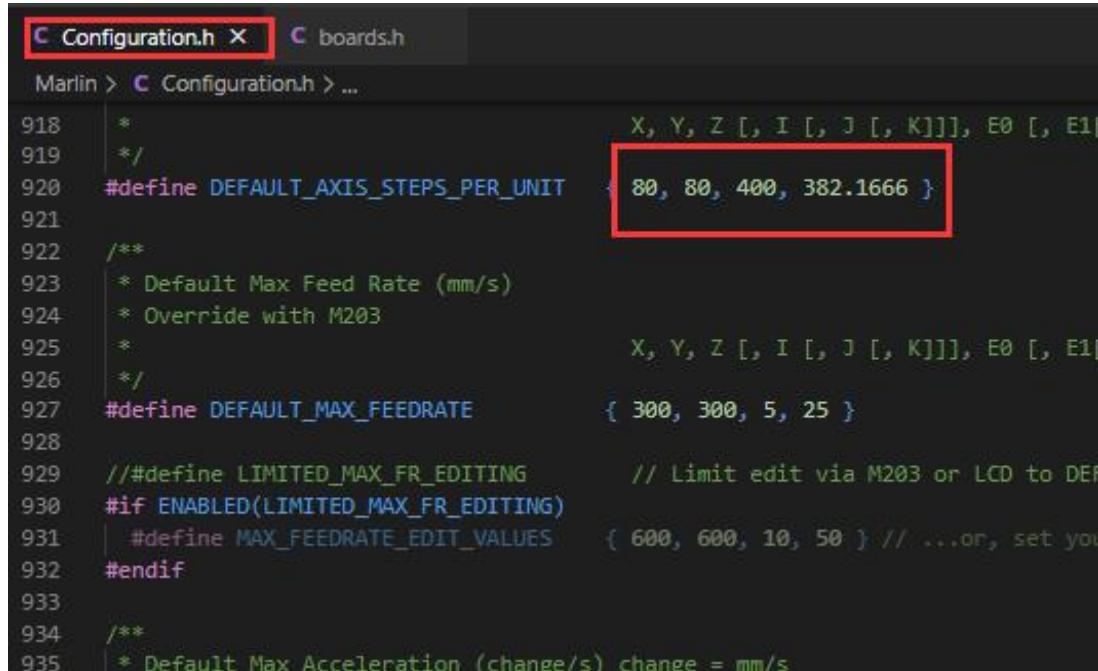
5.2.4 Pulse setting

#Define DEFAULT_AXIS_STEPS_PER_UNIT {80, 80, 400, 382.1666}

in the configuration file to set the pulses of the X, Y, Z, and E axes

respectively (based on Voron2.4, the default pulses are 80, 80, 400,

382.1666) .



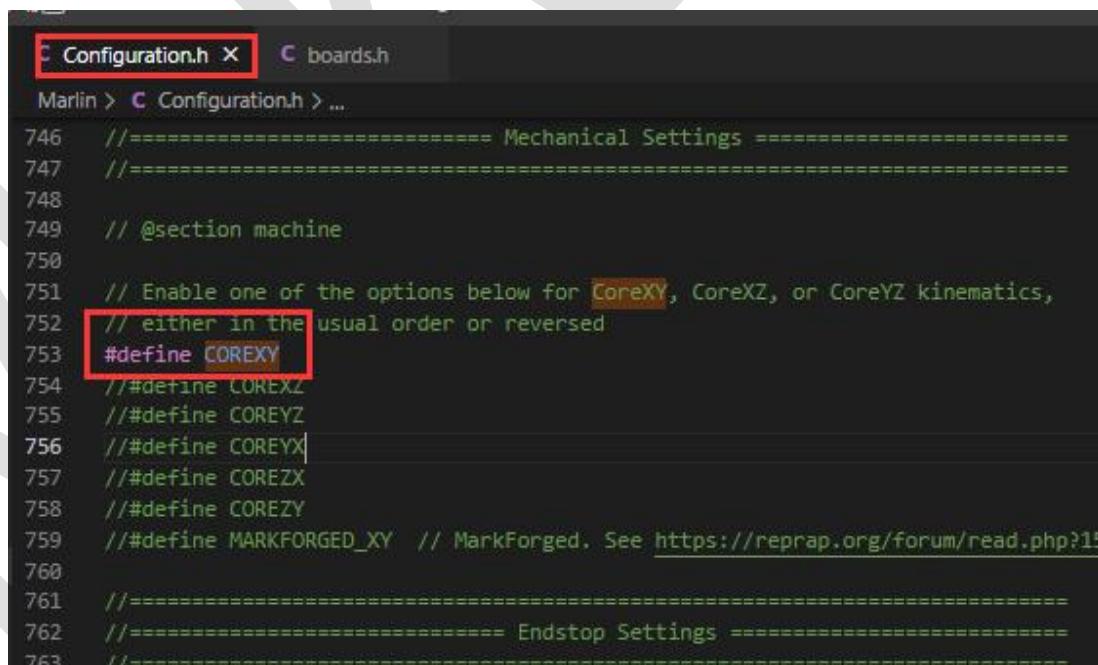
```

Configuration.h X boards.h
Marlin > Configuration.h > ...
918  * X, Y, Z [, I [, J [, K]]], E0 [, E1]
919  */
920 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 382.1666 }
921
922 /**
923 * Default Max Feed Rate (mm/s)
924 * Override with M203
925 *
926 */
927 #define DEFAULT_MAX_FEEDRATE X, Y, Z [, I [, J [, K]]], E0 [, E1]
928
929 // #define LIMITED_MAX_FR_EDITING
930 #if ENABLED(LIMITED_MAX_FR_EDITING)
931   #define MAX_FEEDRATE_EDIT_VALUES { 300, 300, 5, 25 }
932 #endif
933
934 /**
935 * Default Max Acceleration (change/s). change = mm/s

```

5.2.5 Machine type configuration

The machine type of Voron 2.4 is corexy, you need to enable corexy in the configuration file



```

Configuration.h X boards.h
Marlin > Configuration.h > ...
746 //===== Mechanical Settings =====
747 //=====
748 // @section machine
749
750 // Enable one of the options below for CoreXY, CoreXZ, or CoreYZ kinematics,
751 // either in the usual order or reversed
752 #define COREXY
753 // #define COREXZ
754 // #define COREYZ
755 // #define COREYX
756 // #define COREZX
757 // #define COREZY
758 // #define MARKFORGED_XY // MarkForged. See https://reprap.org/forum/read.php?15
759
760 //=====
761 //===== Endstop Settings =====
762 //=====
763 //=====

```

5.2.6 Zero return direction setting

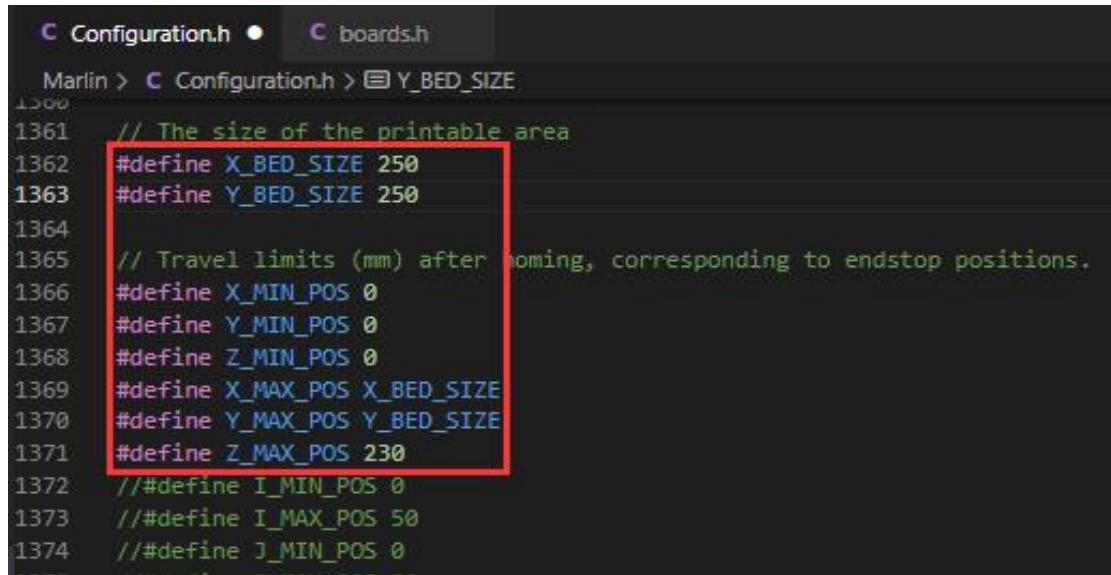
Set the zeroing direction in the configuration file, -1 is the minimum direction, 1 is the maximum direction, based on voron 2.4, the zero point of the X axis and the Y axis is in the upper right corner, then X and Y are zero toward the maximum direction, Z The axis returns to zero in the minimum direction (the default firmware has been configured to return to zero in X and Y to the maximum direction, and Z to zero in the minimum direction. Note that you cannot force the upper right corner to be set to the minimum zero point. Forced setting will cause the X and Y axis printing to appear mirrored).

5.2.7 Print platform range setting

Voron 2.4 machine print size is divided into 3 types (unit mm):

250x250x230, 300x300x280, 350x350x330; you can set according to the

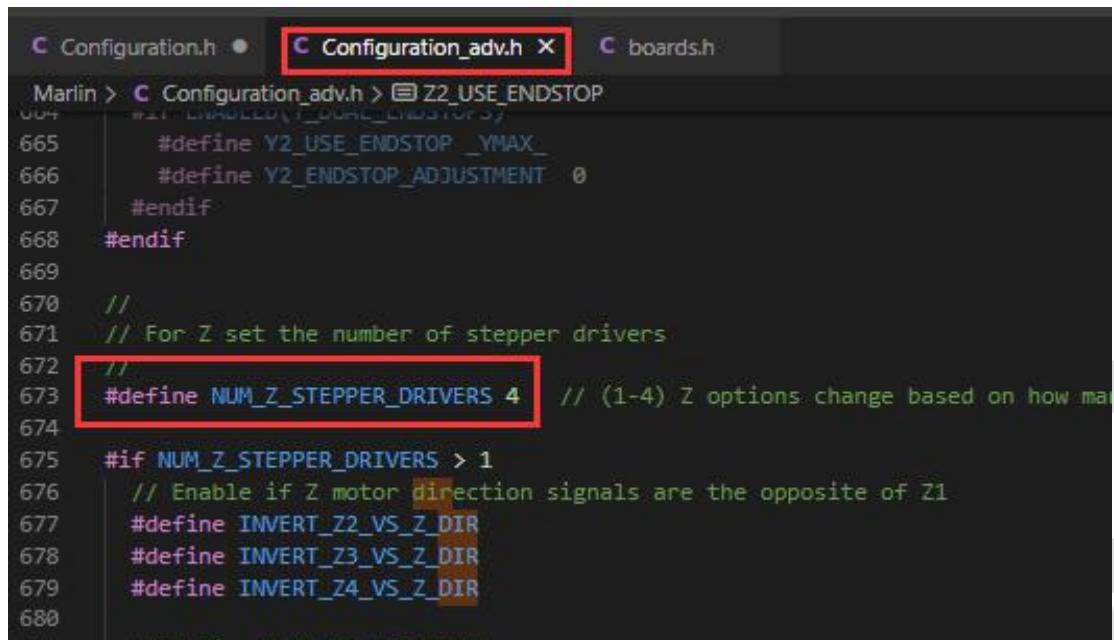
size of your own machine, the default firmware size is 250x250x230.



```
Configuration.h ● boards.h
Marlin > Configuration.h > Y_BED_SIZE
1361 // The size of the printable area
1362 #define X_BED_SIZE 250
1363 #define Y_BED_SIZE 250
1364
1365 // Travel limits (mm) after homing, corresponding to endstop positions.
1366 #define X_MIN_POS 0
1367 #define Y_MIN_POS 0
1368 #define Z_MIN_POS 0
1369 #define X_MAX_POS X_BED_SIZE
1370 #define Y_MAX_POS Y_BED_SIZE
1371 #define Z_MAX_POS 230
1372 // #define I_MIN_POS 0
1373 // #define I_MAX_POS 50
1374 // #define J_MIN_POS 0
1375 // #define J_MAX_POS 50
```

5.2.8 4Z motor configuration

Configure `#define NUM_Z_STEPPER_DRIVERS 1` as `#define NUM_Z_STEPPER_DRIVERS 4` in the advanced configuration file (the default firmware has enabled 4Z motors)



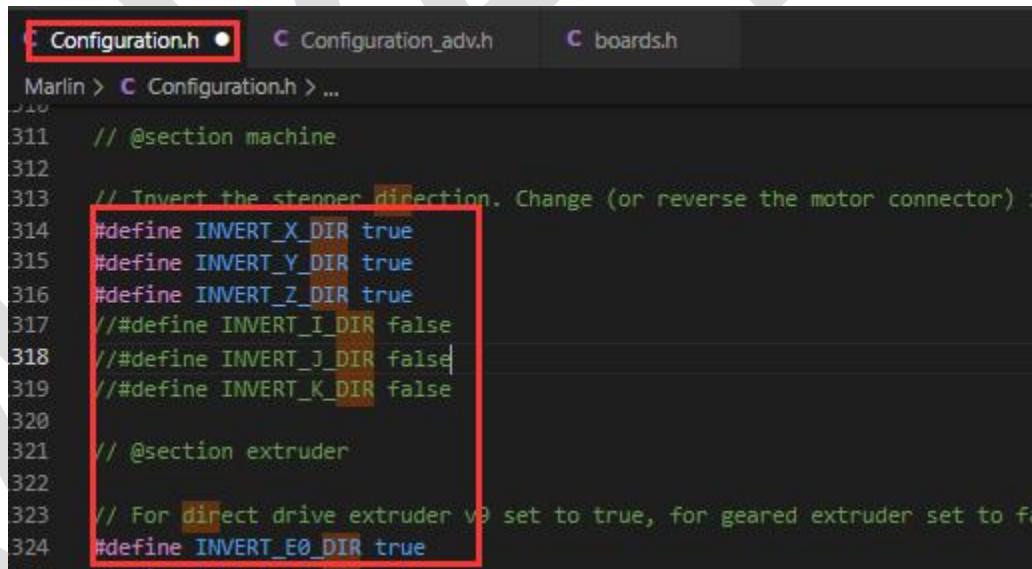
```

C Configuration.h ● C Configuration_adv.h ✘ C boards.h
Marlin > C Configuration_adv.h > Z2_USE_ENDSTOP
665     #define Y2_USE_ENDSTOP _YMAX_
666     #define Y2_ENDSTOP_ADJUSTMENT 0
667     #endif
668 #endif
669
670 /**
671 // For Z set the number of stepper drivers
672 /**
673 #define NUM_Z_STEPPER_DRIVERS 4 // (1-4) Z options change based on how many
674
675 #if NUM_Z_STEPPER_DRIVERS > 1
676     // Enable if Z motor direction signals are the opposite of Z1
677     #define INVERT_Z2_VS_Z_DIR
678     #define INVERT_Z3_VS_Z_DIR
679     #define INVERT_Z4_VS_Z_DIR
680

```

5.2.9 Motor direction setting

The motor direction configuration in the configuration file is as follows:



```

C Configuration.h ● C Configuration.h ...
Marlin > C Configuration.h ...
311 // @section machine
312
313 // Invert the stepper direction. Change (or reverse the motor connector)
314 #define INVERT_X_DIR true
315 #define INVERT_Y_DIR true
316 #define INVERT_Z_DIR true
317 // #define INVERT_I_DIR false
318 // #define INVERT_J_DIR false
319 // #define INVERT_K_DIR false
320
321 // @section extruder
322
323 // For direct drive extruder v0 set to true, for geared extruder set to false
324 #define INVERT_E0_DIR true

```

1) X and Y axis movement direction setting:

Whether the moving direction is correct, we can check by operating the

movement on the screen. Before powering on, we can manually push the extrusion head to the middle of the platform. After powering on, press the rotary encoder to enter the movement→movement axis→movement X Axis/Move the Y axis (be careful not to move too far, otherwise it will hit the extruder). After power-on, the default X and Y axis positions are the maximum positions and can only move to the minimum direction; when looking at the printer, when moving the X axis , The correct direction is to move the X axis to the left; when moving the Y axis, the correct direction of the Y axis is to move forward; if the moving direction is wrong, you need to modify the motor direction in the configuration file.

The modification method and reference are as follows:

Error 1: When the on-screen operation moves the X-axis, the Y-axis on the machine moves forward, and when the on-screen operation moves the Y-axis, the X-axis on the machine moves to the left, it is the A motor (the Y-axis motor moving direction) is wrong, B Motor (the direction of the X-axis motor is correct), you need to configure the Y-axis motor in the opposite direction in the configuration file, then change the original true to false, or change the original false to true;

Error 2: When operating on the screen to move the X axis, the Y axis on the machine moves backward, and when the on screen operation moves the Y axis, the X axis on the machine moves to the right, it is the B motor

(X axis motor moving direction) is wrong, A Motor (Y-axis motor is moving in the correct direction), you need to configure the X-axis motor in the opposite direction in the configuration file, then change the original true to false, or change the original false to true;

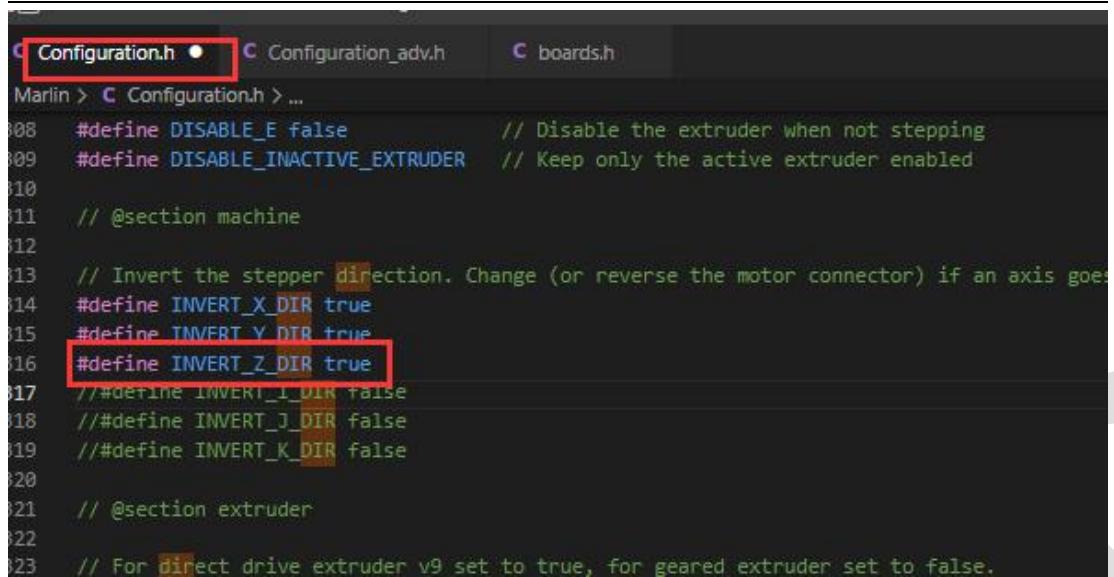
Error 3: When operating on the screen to move the X-axis, the X-axis on the machine moves to the right, and when the on-screen operation moves the Y-axis, the Y-axis on the machine moves backward. The directions of the A and B motors are both wrong. In the configuration file, if the movement directions of the X and Y axis motors are reversed, the original true is changed to false, or the original false is changed to true;

2)Z-axis motor direction configuration:

The Voron 2.4 machine has 4 Z-axis motors, and the configuration of the motor direction is as follows:

First, you need to confirm the direction of the Z1 motor. Operate on the screen to move the Z axis up and down (be careful not to move too far).

The Z1 motor can move up and down according to the operation correctly. If it is not correct, you need to modify the Z1 motor. Direction, change the original false to true and true to false.



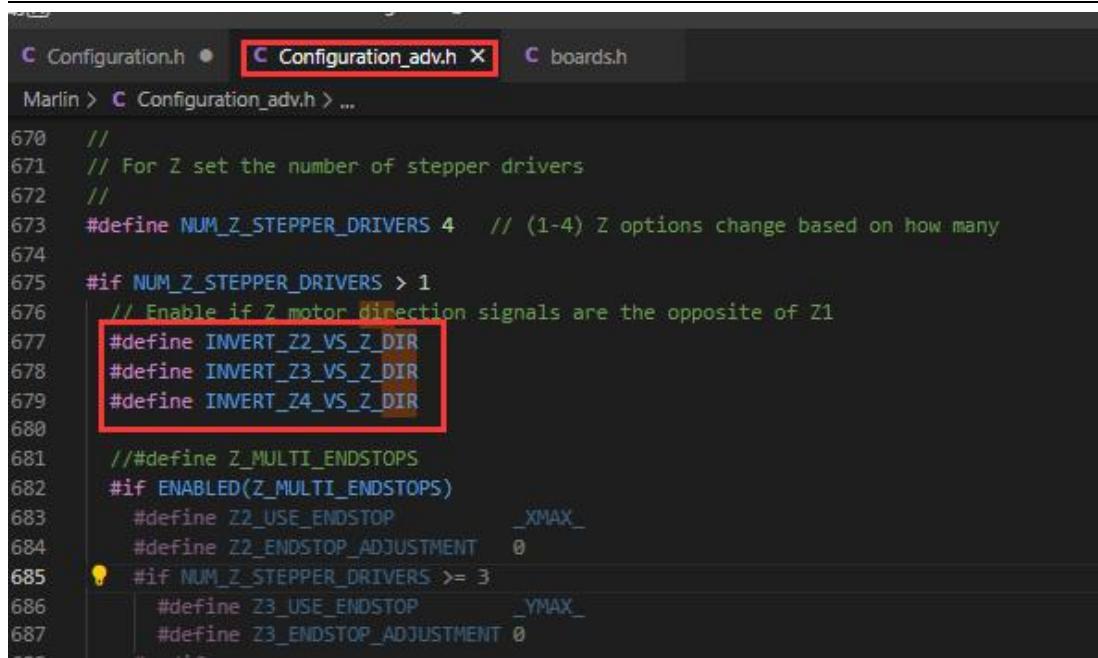
```
Configuration.h • Configuration_adv.h boards.h

Marlin > Configuration.h > ...

808 #define DISABLE_E false          // Disable the extruder when not stepping
809 #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
810
811 // @section machine
812
813 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes:
814 #define INVERT_X_DIR true
815 #define INVERT_Y_DIR true
816 #define INVERT_Z_DIR true
817 // #define INVERT_I_DIR false
818 // #define INVERT_J_DIR false
819 // #define INVERT_K_DIR false
820
821 // @section extruder
822
823 // For direct drive extruder v9 set to true, for geared extruder set to false.
```

To modify the direction of the Z2, Z3, and Z4 motors, the same is to move the Z axis on the screen, and then observe whether the moving directions of the Z2, Z3, and Z4 motors can be up and down according to the correct operation. If the direction is not correct, you need to enter the advanced configuration file Configure in;

Enabling the Z2, Z3, Z4 motor direction to be consistent with Z1 will change the Z2, Z3, Z4 motor direction respectively; if it is enabled by default, if you need to modify the motor direction, you need to shield.



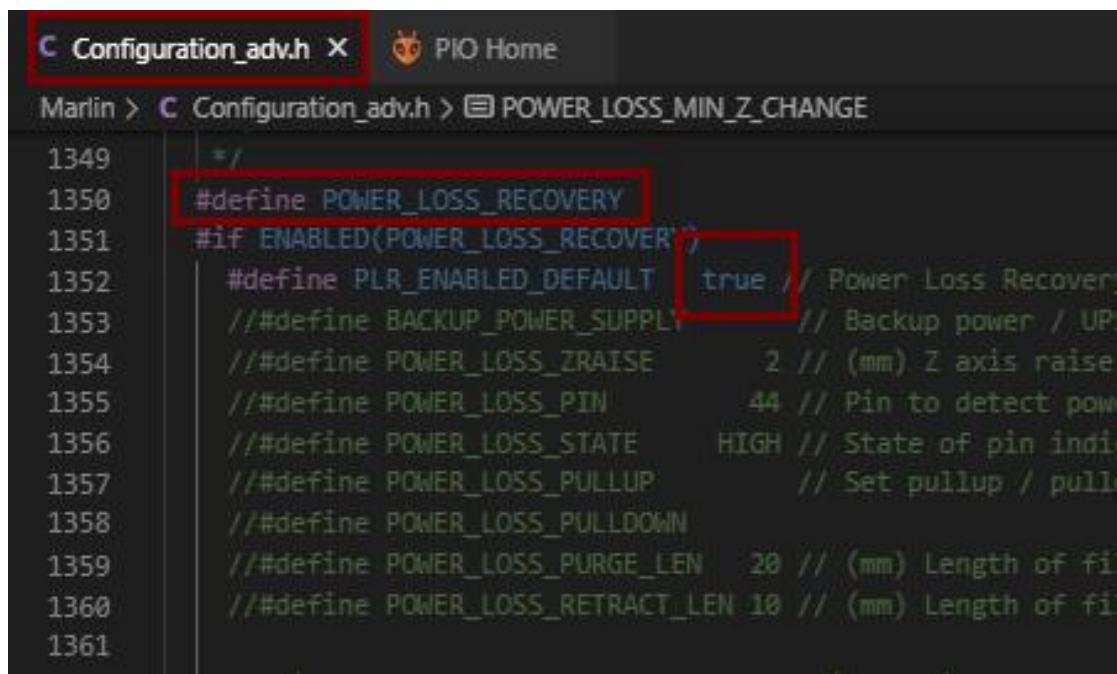
```
670 //  
671 // For Z set the number of stepper drivers  
672 //  
673 #define NUM_Z_STEPPER_DRIVERS 4 // (1-4) Z options change based on how many  
674  
675 #if NUM_Z_STEPPER_DRIVERS > 1  
676     // Enable if Z motor direction signals are the opposite of Z1  
677     #define INVERT_Z2_VS_Z_DIR  
678     #define INVERT_Z3_VS_Z_DIR  
679     #define INVERT_Z4_VS_Z_DIR  
680  
681 // #define Z_MULTI_ENDSTOPS  
682 #if ENABLED(Z_MULTI_ENDSTOPS)  
683     #define Z2_USE_ENDSTOP _XMAX_  
684     #define Z2_ENDSTOP_ADJUSTMENT 0  
685     #if NUM_Z_STEPPER_DRIVERS >= 3  
686         #define Z3_USE_ENDSTOP _YMAX_  
687         #define Z3_ENDSTOP_ADJUSTMENT 0  
688     #endif
```

6. Advanced configuration

6.1 Resume after power off (default firmware has been enabled)

In the advanced configuration configuration_adv.h file, enable the following configuration items:

```
#define POWER_LOSS_RECOVERY  
#define PLR_ENABLED_DEFAULT true
```



```
Marlin > C Configuration_adv.h > POWER LOSS MIN Z CHANGE
1349 */
1350 #define POWER LOSS RECOVERY
1351 #if ENABLED(POWER LOSS RECOVERY)
1352     #define PLR_ENABLED_DEFAULT true // Power Loss Recovery
1353     //#define BACKUP_POWER_SUPPLY // Backup power / UPS
1354     //#define POWER LOSS_ZRAISE 2 // (mm) Z axis raise
1355     //#define POWER LOSS_PIN 44 // Pin to detect power loss
1356     //#define POWER LOSS_STATE HIGH // State of pin indicates power loss
1357     //#define POWER LOSS_PULLUP // Set pullup / pulldown
1358     //#define POWER LOSS_PULLDOWN
1359     //#define POWER LOSS_PURGE_LEN 20 // (mm) Length of filament to purge
1360     //#define POWER LOSS_RETRACT_LEN 10 // (mm) Length of filament to retract
1361
```

6.2 Filament of runout sensor

Enable #define FILAMENT_RUNOUT_SENSOR in the configuration file

C Configuration.h M • **C pins_MKS_ROBIN_E3_common.h**

```

Marlin > C Configuration.h > FILAMENT_RUNOUT_SENSOR
1274 * 2. The Print Job Timer has been started with M75.
1275 * 3. The heaters were turned on and PRINTJOB_TIMER_AUTOSTART
1276 *
1277 * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
1278 * For other boards you may need to define FIL_RUNOUT_PIN, FIL_
1279 */
1280 #define FILAMENT_RUNOUT_SENSOR
1281 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
1282     #define FIL_RUNOUT_ENABLED_DEFAULT true // Enable the sensor
1283     #define NUM_RUNOUT_SENSORS 1           // Number of sensors
1284
1285     #define FIL_RUNOUT_STATE LOW          // Pin state indicating
1286     #define FIL_RUNOUT_PULLUP            // Use internal pullup for f
1287     //##define FIL_RUNOUT_PULLDOWN      // Use internal pulldown for
1288     //##define WATCH_ALL_RUNOUT_SENSORS // Execute runout script on
1289     // This is automatically enabled
1290
1291 // Override individually if the runout sensors vary
1292 //##define FIL_RUNOUT1_STATE LOW
1293 //##define FIL_RUNOUT1_PULLUP
1294 //##define FIL_RUNOUT1_PULLDOWN

```

Set the sensor level in the configuration file (LOW/HIGH)

C Configuration.h M

```

Marlin > C Configuration.h
1276 *
1277 * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
1278 * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_
1279 */
1280 #define FILAMENT_RUNOUT_SENSOR
1281 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
1282     #define FIL_RUNOUT_ENABLED_DEFAULT true // Enable the sensor on start
1283     #define NUM_RUNOUT_SENSORS 1           // Number of sensors, up to 2
1284
1285     #define FIL_RUNOUT_STATE LOW          // Pin state indicating that
1286     #define FIL_RUNOUT_PULLUP            // Use internal pullup for f
1287     //##define FIL_RUNOUT_PULLDOWN      // Use internal pulldown for
1288     //##define WATCH_ALL_RUNOUT_SENSORS // Execute runout script on
1289     // This is automatically enabled
1290
1291 // Override individually if the runout sensors vary
1292 //##define FIL_RUNOUT1_STATE LOW
1293 //##define FIL_RUNOUT1_PULLUP
1294 //##define FIL_RUNOUT1_PULLDOWN

```

Enable `#define NOZZLE_PARK_FEATURE` in the configuration file

C Configuration.h M C pins_MKS_ROBIN_E3_common.h

```

Marlin > C Configuration.h > ...
1717 * P0 (Default) If Z is below park Z raise the nozzle.
1718 * P1 Raise the nozzle always to Z-park height.
1719 * P2 Raise the nozzle by Z-park amount, limited to Z_MAX
1720 */
1721 #define NOZZLE_PARK_FEATURE
1722
1723 #if ENABLED(NOZZLE_PARK_FEATURE)
1724     // Specify a park position as { X, Y, Z_raise }
1725     #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), Z_RAISE }
1726     // #define NOZZLE_PARK_X_ONLY           // X move only is required
1727     // #define NOZZLE_PARK_Y_ONLY           // Y move only is required
1728     #define NOZZLE_PARK_Z_RAISE_MIN    2 // (mm) Always raise Z at least this much
1729     #define NOZZLE_PARK_XY_FEEDRATE 100 // (mm/s) X and Y axis feedrate
1730     #define NOZZLE_PARK_Z_FEEDRATE   5 // (mm/s) Z axis feedrate
1731 #endif
1732
1733 /**
1734 * Clean Nozzle Feature -- EXPERIMENTAL
1735 *
1736 * Adds the G12 command to perform a nozzle cleaning process
1737 */

```

Enable #define ADVANCED_PAUSE_FEATURE in the advanced configuration file

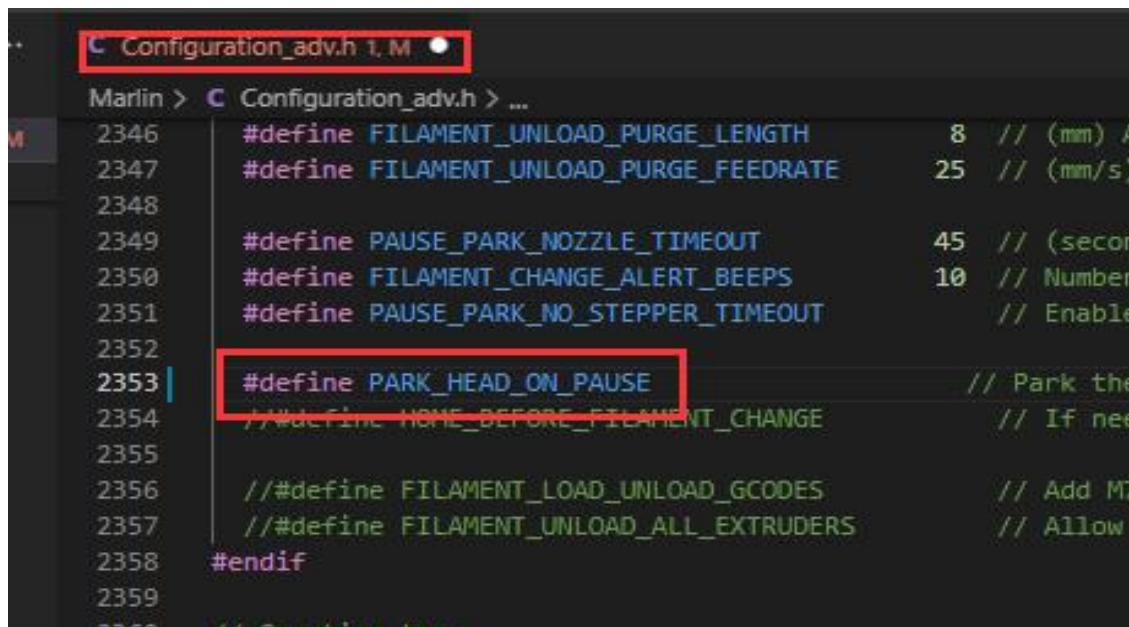
C Configuration_adv.h M C Configuration_adv.h > ADVANCED_PAUSE_FEATURE

```

Marlin > C Configuration_adv.h > ADVANCED_PAUSE_FEATURE
2312 * Requires NOZZLE_PARK_FEATURE.
2313 * This feature is required for the default FILAMENT_RUNOUT_SCRIPT.
2314 */
2315 #define ADVANCED_PAUSE_FEATURE
2316 #if ENABLED(ADVANCED_PAUSE_FEATURE)
2317     #define PAUSE_PARK_RETRACT_FEEDRATE      60 // (mm/s) In
2318     #define PAUSE_PARK_RETRACT_LENGTH        2 // (mm) Init
2319     // This short length prevents filament from
2320     #define FILAMENT_CHANGE_UNLOAD_FEEDRATE 10 // (mm/s) Un
2321     #define FILAMENT_CHANGE_UNLOAD_ACCEL   25 // (mm/s^2)
2322     #define FILAMENT_CHANGE_UNLOAD_LENGTH  100 // (mm) The
2323                                         // For Bowden
2324                                         // For direct
2325                                         // Set to
2326     #define FILAMENT_CHANGE_SLOW_LOAD_FEEDRATE 6 // (mm/s) Sl
2327     #define FILAMENT_CHANGE_SLOW_LOAD_LENGTH  0 // (mm) Sto

```

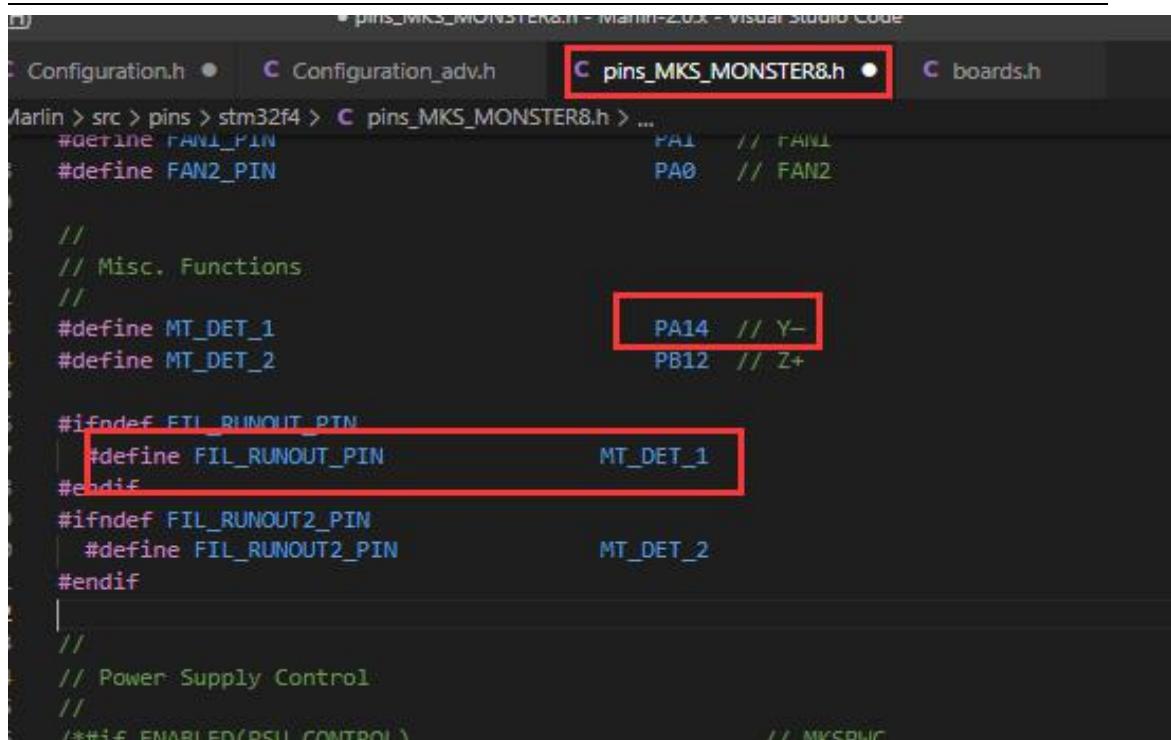
Enable #define PARK_HEAD_ON_PAUSE in the advanced configuration file



```
Marlin > C Configuration_adv.h ...  
2346 #define FILAMENT_UNLOAD_PURGE_LENGTH 8 // (mm) A  
2347 #define FILAMENT_UNLOAD_PURGE_FEEDRATE 25 // (mm/s)  
2348  
2349 #define PAUSE_PARK_NOZZLE_TIMEOUT 45 // (second)  
2350 #define FILAMENT_CHANGE_ALERT_BEEPS 10 // Number of beeps  
2351 #define PAUSE_PARK_NO_STEPPER_TIMEOUT // Enable  
2352  
2353 #define PARK_HEAD_ON_PAUSE // Park the head on pause  
2354 // #define HOME_BEFORE_FILAMENT_CHANGE // If needed  
2355  
2356 // #define FILAMENT_LOAD_UNLOAD_GCODES // Add M7  
2357 // #define FILAMENT_UNLOAD_ALL_EXTRUDERS // Allow  
2358 #endif  
2359 // Configuration_adv.h
```

Add the material break detection pin in the pins_MONSTER8.h file

(Note: This pin port needs to be configured according to the pin connected to the material break detection switch. Take the material break detection connected to the limit of X_MIN as an example, the pin of the X_MIN limit is PA14)



```
Configuration.h • Configuration_adv.h • pins_MKS_MONSTER8.h • boards.h
Marlin > src > pins > stm32f4 > pins_MKS_MONSTER8.h > ...
#define FAN1_PIN PA1 // FAN1
#define FAN2_PIN PA0 // FAN2

// Misc. Functions
//
#define MT_DET_1
#define MT_DET_2

#ifndef FIL_RUNOUT_PIN
#define FIL_RUNOUT_PIN MT_DET_1
#endif

#ifndef FIL_RUNOUT2_PIN
#define FIL_RUNOUT2_PIN MT_DET_2
#endif

// Power Supply Control
// /*#if ENABLED(PSU_CONTROL)
//   // MKSPLIC
```

6.3 TMC 2208/TMC 2209/TMC 2225/TMC 2226 uart mode

Enable the TMC driver in the configuration file, TMC2208 and TMC2225 are both enabled as TMC 2208, TMC 2209 and TMC2226 are both enabled as TMC2209, (based on Voron 2.4, the default firmware has enabled TMC2209 UART mode)

Marlin > Configuration.h > ...

```
852     * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6
853     */
854     #define X_DRIVER_TYPE    TMC2209
855     #define Y_DRIVER_TYPE    TMC2209
856     #define Z_DRIVER_TYPE    TMC2209
857     //#define X2_DRIVER_TYPE A4988
858     //#define Y2_DRIVER_TYPE A4988
859     #define Z2_DRIVER_TYPE   TMC2209
860     #define Z3_DRIVER_TYPE   TMC2209
861     #define Z4_DRIVER_TYPE   TMC2209
862     //#define I_DRIVER_TYPE  A4988
863     //#define J_DRIVER_TYPE  A4988
864     //#define K_DRIVER_TYPE  A4988
865     #define E0_DRIVER_TYPE   TMC2209
866     //#define E1_DRIVER_TYPE A4988
867     //#define E2_DRIVER_TYPE A4988
868     //#define E3_DRIVER_TYPE A4988
869     //#define E4_DRIVER_TYPE A4988
870     //#define E5_DRIVER_TYPE A4988
871     //#define E6_DRIVER_TYPE A4988
```

7. 4Z automatic leveling (sensor is PL08N)

7.1 Set the sensor trigger level

PL08N is a normally open switch, the trigger level is set to true

```
Marlin > Configuration.h • Configuration_adv.h • pins_MKS_MONSTER8.h • boards.h
823 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
824 #define Z_MIN_ENDSTOP_INVERTING true// Set to true to invert the logic of the endstop.
825 #define I_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
826 #define J_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
827 #define K_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
828 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
829 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
830 #define Z_MAX_ENDSTOP_INVERTING true// Set to true to invert the logic of the endstop.
831 #define I_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
832 #define J_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
833 #define K_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
834 #define Z_MIN_PROBE_ENDSTOP_INVERTING true // Set to true to invert the logic of the probe.
835
836 /**
837 * Stepper Drivers
838 *
839 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
840 * stepper drivers that support them. You may also override timing options in Configuration_adv.h
841 *
```

7.2 Set sensor signal pin

The sensor signal pin needs to be set according to the wiring of the motherboard (the black line is the signal line of PL08N, which is connected to the Z_MAX limit PB12 based on Voron 2.4).

Note: Use z_safe_homin function, skip this configuration, see 7.8 z_safe_homing configuration (default firmware is to enable PL08N z_safe_homin function)

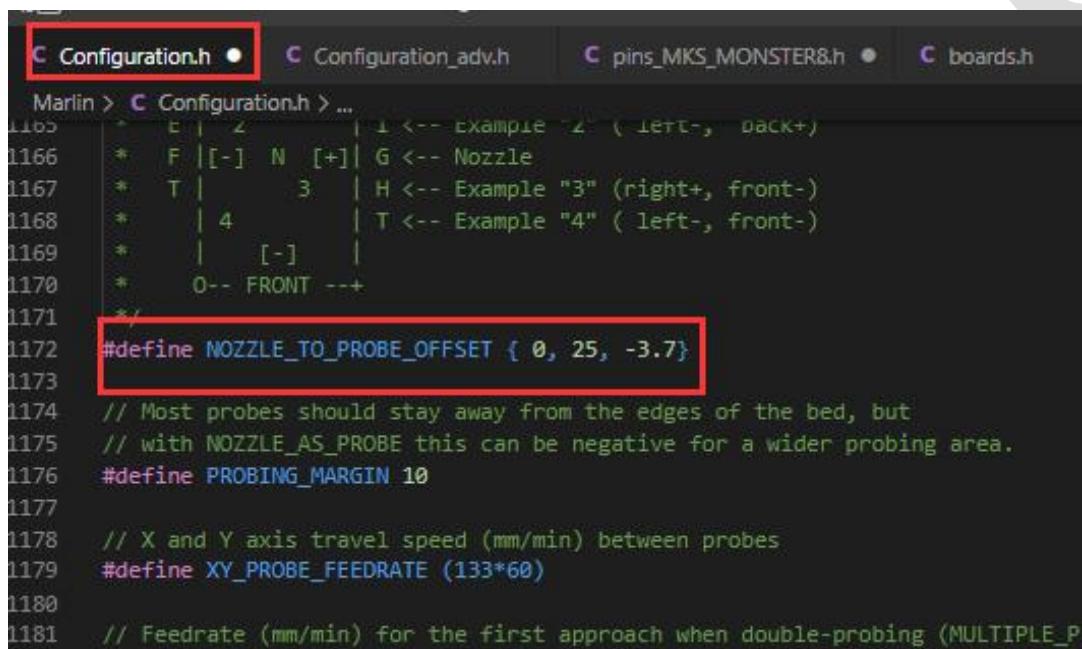
Configuration.h • PIO Home pins.h pins_MKS_ESP_NANO

Marlin > Configuration.h > Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN

```
903 /**
904  * Enable this option for a probe connected to the Z-MIN pin.
905  * The probe replaces the Z-MIN endstop and is used for Z homing.
906  * (Automatically enables USE_PROBE_FOR_Z_HOMING.)
907 */
908 // #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
909 // Force the use of the probe for Z-axis homing
910 // #define USE_PROBE_FOR_Z_HOMING
911 // disnable this item
912 /**
913  * Z_MIN_PROBE_PIN
914  *
915  * Define this pin if the probe is not connected to Z_MIN_PIN.
916  * If not defined the default pin for the selected MOTHERBOARD
917  * will be used. Most of the time the default is what you want.
918  *
919  * - The simplest option is to use a free endstop connector.
920  * - Use 5V for powered (usually inductive) sensors.
921  *
922  * - RAMPS 1.3/1.4 boards may use the 5V, GND, and Aux4->D32 pins.
923  *   - For simple switches connect...
924  *     - normally-closed switches to GND and D32.
925  *     - normally-open switches to 5V and D32.
926  */
927 // enable this item
928 // #define Z_MIN_PROBE_PIN PB12 // Pin 32 is the RAMPS default
929
930 /**
931  * Probe Type
932 */
```

7.3 Offset setting

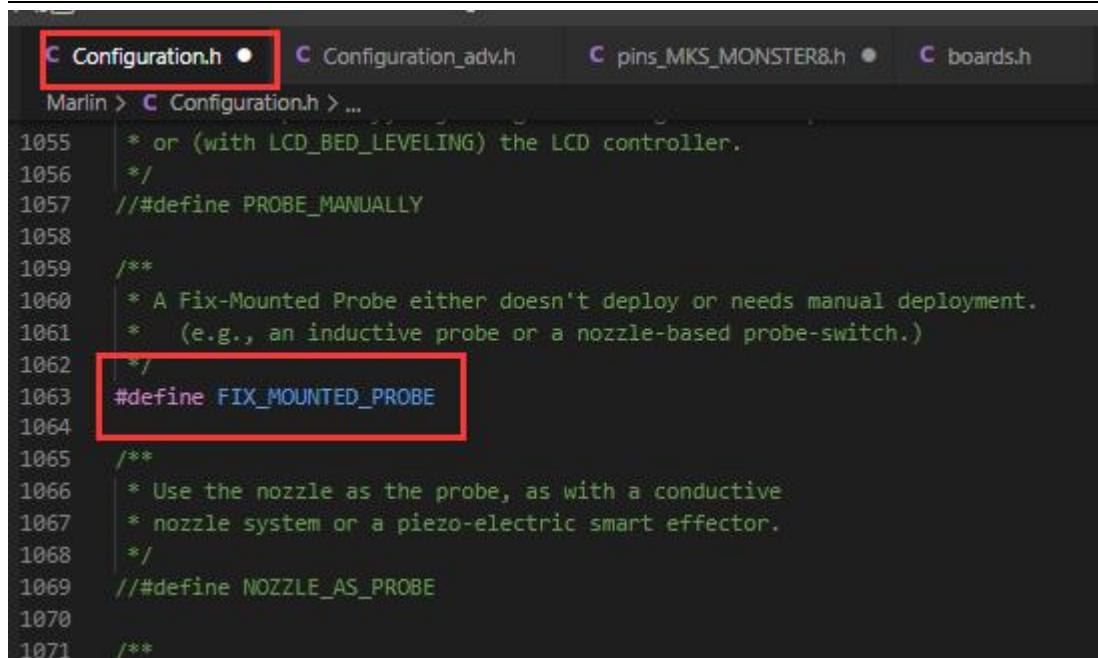
Based on Voron2.4, the X and Y axis offsets are 0 and 25, and the Z offset (z_offset) can be tested and adjusted after leveling.



```
Configuration.h • Configuration_adv.h pins_MKS_MONSTER8.h • boards.h
Marlin > Configuration.h > ...
1165   * E | 2 | L <-- Example "2" (left-, back+)
1166   * F | [-] N [+]| G <-- Nozzle
1167   * T | 3 | H <-- Example "3" (right+, front-)
1168   * | 4 | T <-- Example "4" ( left-, front-)
1169   * | [-] |
1170   * 0-- FRONT ---+
1171 */
1172 #define NOZZLE_TO_PROBE_OFFSET { 0, 25, -3.7}
1173
1174 // Most probes should stay away from the edges of the bed, but
1175 // with NOZZLE_AS_PROBE this can be negative for a wider probing area.
1176 #define PROBING_MARGIN 10
1177
1178 // X and Y axis travel speed (mm/min) between probes
1179 #define XY_PROBE_FEEDRATE (133*60)
1180
1181 // Feedrate (mm/min) for the first approach when double-probing (MULTIPLE_P
```

7.4 Configure the probe type

The detection type of PL08N is #define FIX_MOUNTED_PROBE (the default firmware is enabled)



```
Marlin > C Configuration.h > ...
1055 * or (with LCD_BED_LEVELING) the LCD controller.
1056 */
1057 //#define PROBE_MANUALLY
1058
1059 /**
1060 * A Fix-Mounted Probe either doesn't deploy or needs manual deployment.
1061 * (e.g., an inductive probe or a nozzle-based probe-switch.)
1062 */
1063 #define FIX_MOUNTED_PROBE
1064
1065 /**
1066 * Use the nozzle as the probe, as with a conductive
1067 * nozzle system or a piezo-electric smart effector.
1068 */
1069 //#define NOZZLE_AS_PROBE
1070
1071 /**
```

7.5 Enable 4Z automatic leveling

Enable 4Z auto-leveling in the advanced configuration file (enabled by default firmware)

```

Configuration_adv.h X pins_MKS_MONSTER8.h boards.h
Marlin > Configuration_adv.h > ...
850  * Add the G34 command to align multiple Z steppers using a bed probe.
851  */
852 #define Z_STEPPER_AUTO_ALIGN
853 #if ENABLED(Z_STEPPER_AUTO_ALIGN)
854  // Define probe X and Y positions for Z1, Z2 [, Z3 [, Z4]]
855  // If not defined, probe limits will be used.
856  // Override with 'M422 S<index> X<pos> Y<pos>'
857  // #define Z_STEPPER_ALIGN_XY { { 10, 190 }, { 100, 10 }, { 190, 190 } }
858
859 /**
860  * Orientation for the automatically-calculated probe positions.
861  * Override Z stepper align points with 'M422 S<index> X<pos> Y<pos>'
862  *
863  * 2 Steppers: (0)      (1)
864  *               |       |
865  *               1     2   2
866  *               |       |
867  *               1       1

```

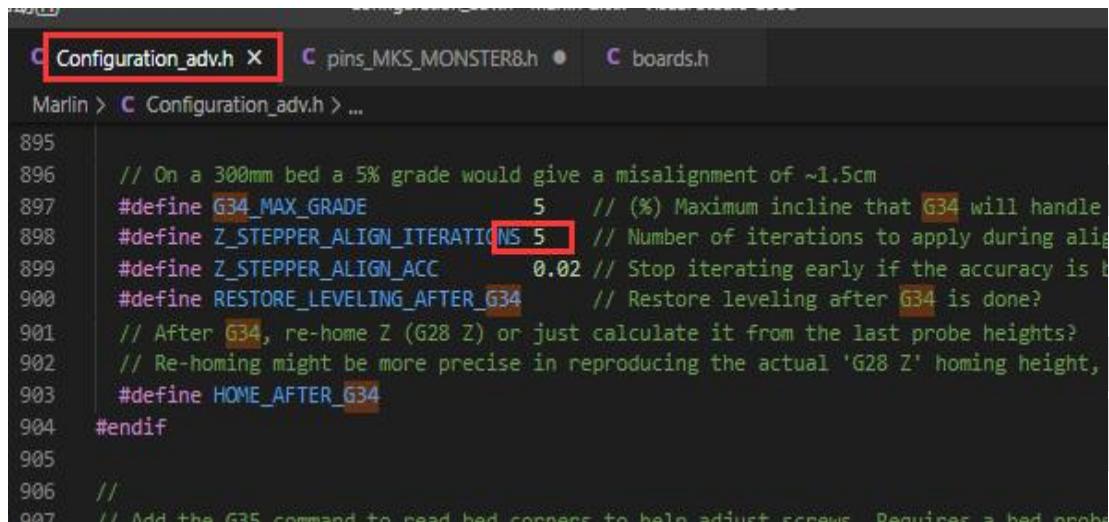
7.6 Set the probe position of 4 motors

```

Configuration_adv.h X pins_MKS_MONSTER8.h boards.h
Marlin > Configuration_adv.h > ...
869  *          3   1   2   1   2   1   2
870  *          |   |   |   |   |   |   |
871  *          1   2   2   3   1   3   1
872  *
873  * 4 Steppers: (0)      (1)      (2)      (3)
874  *          | 4   3   1   4   2   1   3   2
875  *          |          |          |          |
876  *          1   2   2   3   3   4   4   1
877  */
878 #ifndef Z_STEPPER_ALIGN_XY
879 #define Z_STEPPERS_ORIENTATION 0
880#endif
881
882 // Provide Z stepper positions for more rapid convergence in bed alignment
883 // Requires triple stepper drivers (i.e., set NUM_Z_STEPPER_DRIVERS to 3)
884 // #define Z_STEPPER_ALIGN_KNOWN_STEPPER_POSITIONS
885 #if ENABLED(Z_STEPPER_ALIGN_KNOWN_STEPPER_POSITIONS)
886  // Define Stepper XY positions for Z1, Z2, Z3 corresponding to

```

7.7 Set the number of leveling



```
Configuration_adv.h X pins_MKS_MONSTER8.h boards.h
Marlin > Configuration_adv.h ...
895
896 // On a 300mm bed a 5% grade would give a misalignment of ~1.5cm
897 #define G34_MAX_GRADE      5      // (%) Maximum incline that G34 will handle
898 #define Z_STEPPER_ALIGN_ITERATIONS 5 // Number of iterations to apply during align
899 #define Z_STEPPER_ALIGN_ACC    0.02 // Stop iterating early if the accuracy is b
900 #define RESTORE_LEVELING_AFTER_G34 // Restore leveling after G34 is done?
901 // After G34, re-home Z (G28 Z) or just calculate it from the last probe heights?
902 // Re-homing might be more precise in reproducing the actual 'G28 Z' homing height,
903 #define HOME_AFTER_G34
904 #endif
905
906 //
907 // Add the G35 command to read bed sensors to help adjust screws. Requires a bed probe.
```

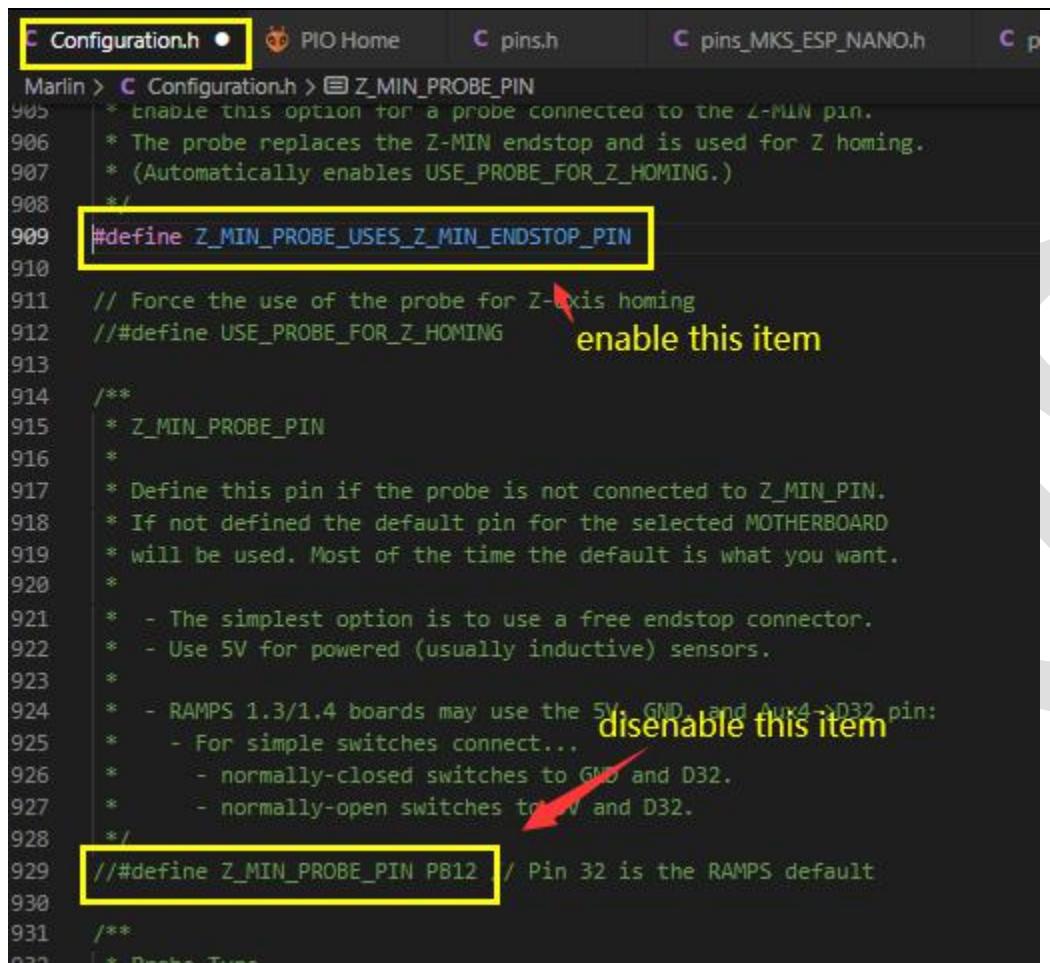
7.8 z_safe_homing configuration

Note: z_safe_homing uses PL08N as the zero return limit of the Z axis.

Other configuration items are the same as those without z_safe_homing (except for setting sensor signal pin in 7.2). If z_safe_homing is not used, the following configuration is not necessary.

- 1) , sensor signal pin setting, use z_safe_homin function, use z_min limit pin, you need to ensure that the configuration file is enabled

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```



```
Marlin > Configuration.h > Z_MIN_PROBE_PIN
905 * enable this option for a probe connected to the Z-MIN pin.
906 * The probe replaces the Z-MIN endstop and is used for Z homing.
907 * (Automatically enables USE_PROBE_FOR_Z_HOMING.)
908 */
909 #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
910 // Force the use of the probe for Z-axis homing
911 //#define USE_PROBE_FOR_Z_HOMING      enable this item
912
913 /**
914 * Z_MIN_PROBE_PIN
915 *
916 * Define this pin if the probe is not connected to Z_MIN_PIN.
917 * If not defined the default pin for the selected MOTHERBOARD
918 * will be used. Most of the time the default is what you want.
919 *
920 * - The simplest option is to use a free endstop connector.
921 * - Use 5V for powered (usually inductive) sensors.
922 *
923 * - RAMPS 1.3/1.4 boards may use the 5V, GND, and Aux4->D32 pin:
924 *   - For simple switches connect...
925 *     - normally-closed switches to GND and D32.
926 *     - normally-open switches to 5V and D32.
927 */
928
929 //#define Z_MIN_PROBE_PIN PB12 // Pin 32 is the RAMPS default
930 /**
931 */
932 /* Probe Type
```

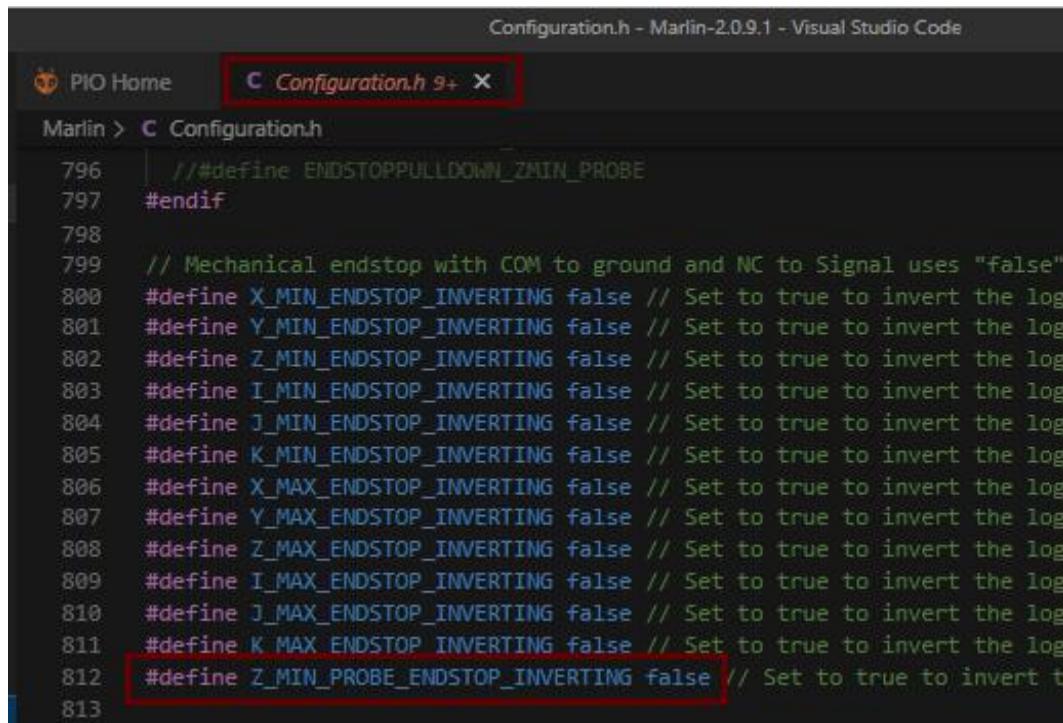
2)enable z_safe_homing in the configuration file

```
Marlin > Configuration.h > ...
1701 // Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
1702 //
1703 // With this feature enabled:
1704 //
1705 // - Allow Z homing only after X and Y homing AND stepper drivers still enable
1706 // - If stepper drivers time out, it will need X and Y homing again before Z homing.
1707 // - Move the Z probe (or nozzle) to a defined XY point before Z Homing.
1708 // - Prevent Z homing when the Z probe is outside bed area.
1709 //
1710 #define Z_SAFE_HOMING
1711
1712 #if ENABLED(Z_SAFE_HOMING)
1713     #define Z_SAFE_HOMING_X_POINT X_CENTER // X point for Z homing
1714     #define Z_SAFE_HOMING_Y_POINT Y_CENTER // Y point for Z homing
1715 #endif
1716
1717 // Homing speeds (mm/min)
1718 #define HOMING_FEEDRATE_MM_M { (50*60), (50*60), (4*60) }
1719
1720 // Validate that endstops are triggered on homing moves
1721 #define VALIDATE_HOMING_ENDSTOPS
1722
1723 // @section calibrate
1724
```

8.4Z automatic leveling (sensor is 3Dtouch)

8.1 Set the sensor trigger level

The level of 3dtouch is set to false

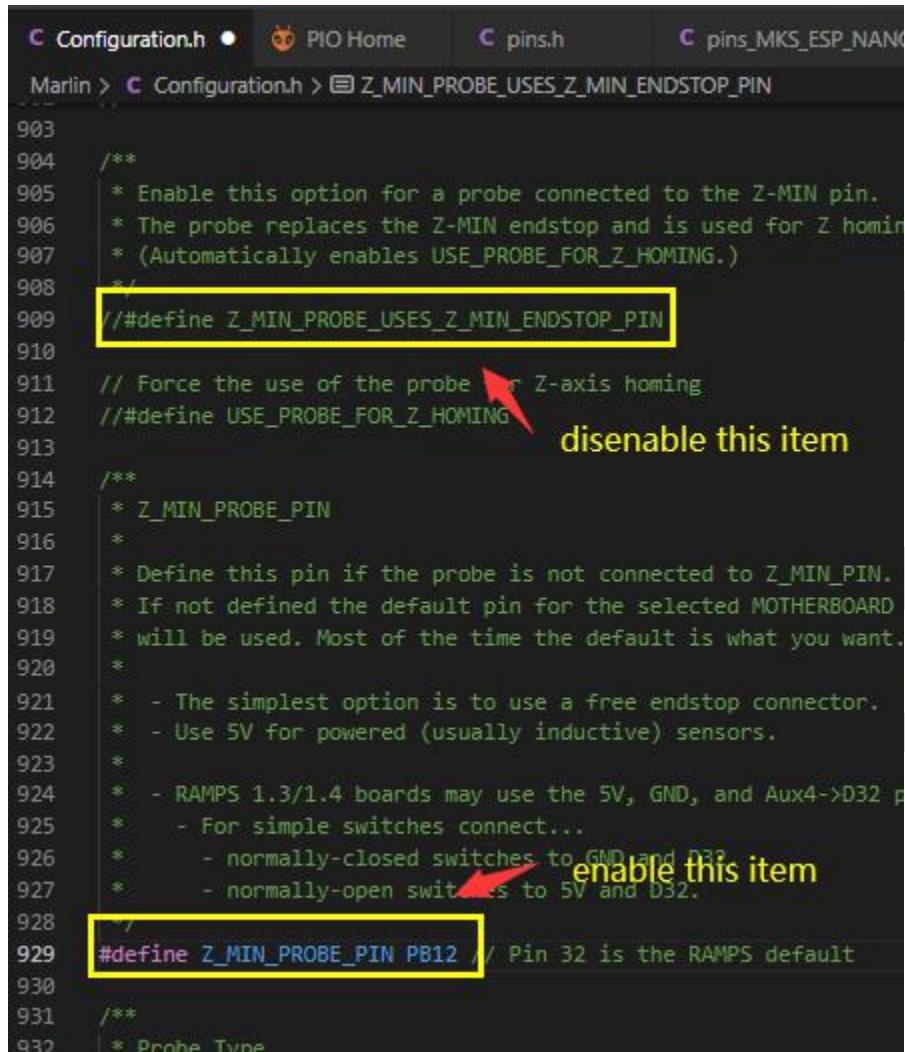


```
Configuration.h - Marlin-2.0.9.1 - Visual Studio Code
PIO Home C Configuration.h 9+ X
Marlin > C Configuration.h
796 // #define ENDSTOPPULLDOWN_ZMIN_PROBE
797 #endif
798
799 // Mechanical endstop with COM to ground and NC to Signal uses "false"
800 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the log
801 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the log
802 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the log
803 #define I_MIN_ENDSTOP_INVERTING false // Set to true to invert the log
804 #define J_MIN_ENDSTOP_INVERTING false // Set to true to invert the log
805 #define K_MIN_ENDSTOP_INVERTING false // Set to true to invert the log
806 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the log
807 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the log
808 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the log
809 #define I_MAX_ENDSTOP_INVERTING false // Set to true to invert the log
810 #define J_MAX_ENDSTOP_INVERTING false // Set to true to invert the log
811 #define K_MAX_ENDSTOP_INVERTING false // Set to true to invert the log
812 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert t
813
```

8.2 Set sensor signal pin

The sensor signal pin needs to be set according to the wiring of the motherboard, and is connected to the Z_MAX limit based on voron2.4
(Note: The signal control line of 3Dtouch is the 2pin line, the black line and the white line, and the white line is the signal line to connect to the Z_MAX limit. Take for example PB12).

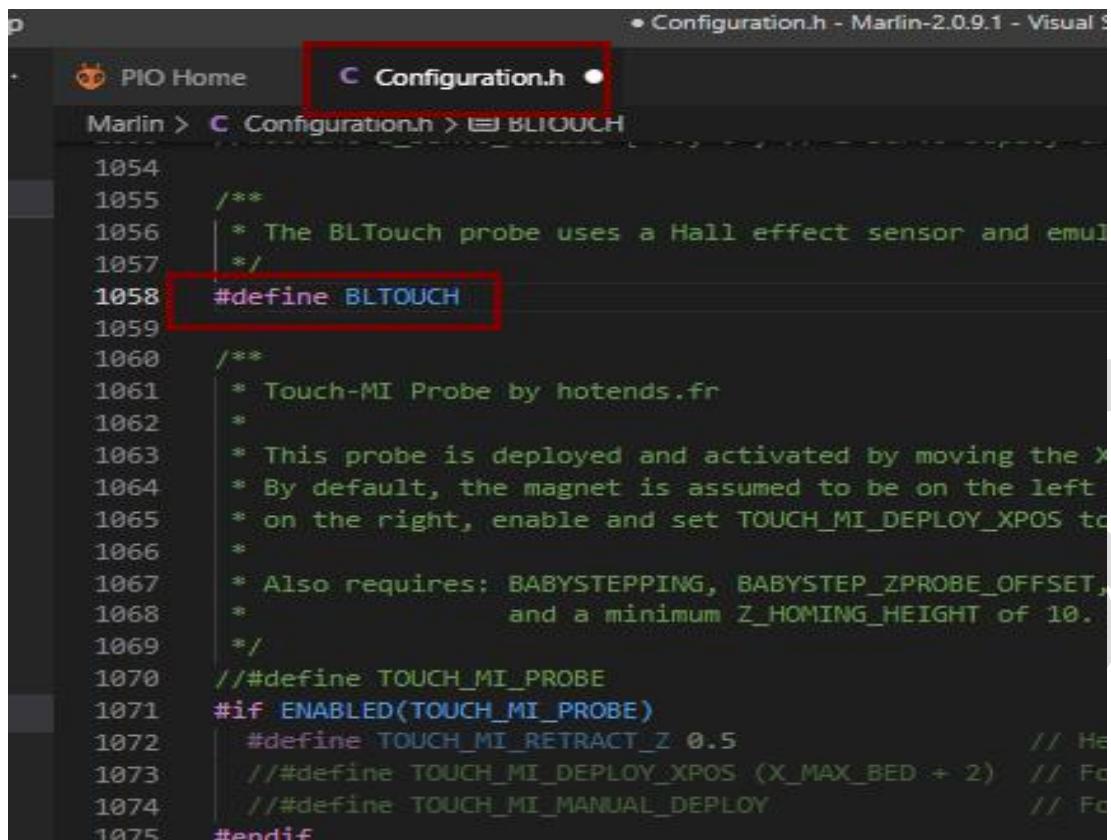
Note: Use z_safe_homin function, skip this configuration, see 8.10 z_safe_homing configuration for details



The screenshot shows a code editor with the file `Configuration.h` open. The code is related to Z-MIN probe configuration. A red arrow points to the line `//#define USE_PROBE_FOR_Z_HOMING` with the annotation "disable this item". Another red arrow points to the line `#define Z_MIN_PROBE_PIN PB12` with the annotation "enable this item".

```
903 /**
904  * Enable this option for a probe connected to the Z-MIN pin.
905  * The probe replaces the Z-MIN endstop and is used for Z homing.
906  * (Automatically enables USE_PROBE_FOR_Z_HOMING.)
907 */
909 // #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
910
911 // Force the use of the probe for Z-axis homing
912 // #define USE_PROBE_FOR_Z_HOMING
913
914 /**
915  * Z_MIN_PROBE_PIN
916  *
917  * Define this pin if the probe is not connected to Z_MIN_PIN.
918  * If not defined the default pin for the selected MOTHERBOARD
919  * will be used. Most of the time the default is what you want.
920  *
921  * - The simplest option is to use a free endstop connector.
922  * - Use 5V for powered (usually inductive) sensors.
923  *
924  * - RAMPS 1.3/1.4 boards may use the 5V, GND, and Aux4->D32 pins
925  *   - For simple switches connect...
926  *     - normally-closed switches to GND and D32
927  *     - normally-open switches to 5V and D32.
928 */
929 #define Z_MIN_PROBE_PIN PB12 // Pin 32 is the RAMPS default
930
931 /**
932  * Probe Type
```

8.3 Configure the probe type (enable BLTOUCH)



```
• Configuration.h - Marlin-2.0.9.1 - Visual Studio Code
  PIO Home Configuration.h
Marlin > C Configuration.h > BLTOUCH
1054
1055  /**
1056  * The BLTouch probe uses a Hall effect sensor and emulates
1057  */
1058 #define BLTOUCH
1059
1060 /**
1061 * Touch-MI Probe by hotends.fr
1062 *
1063 * This probe is deployed and activated by moving the X
1064 * By default, the magnet is assumed to be on the left
1065 * on the right, enable and set TOUCH_MI_DEPLOY_XPOS to
1066 *
1067 * Also requires: BABYSTEPPING, BABYSTEP_ZPROBE_OFFSET,
1068 * and a minimum Z_HOMING_HEIGHT of 10.
1069 */
1070 //#define TOUCH_MI_PROBE
1071 #if ENABLED(TOUCH_MI_PROBE)
1072   #define TOUCH_MI_RETRACT_Z 0.5           // Has
1073   // #define TOUCH_MI_DEPLOY_XPOS (X_MAX_BED+ 2) // For
1074   // #define TOUCH_MI_MANUAL_DEPLOY        // For
1075 #endif
```

8.4 Set the offset between the probe and the extrusion head

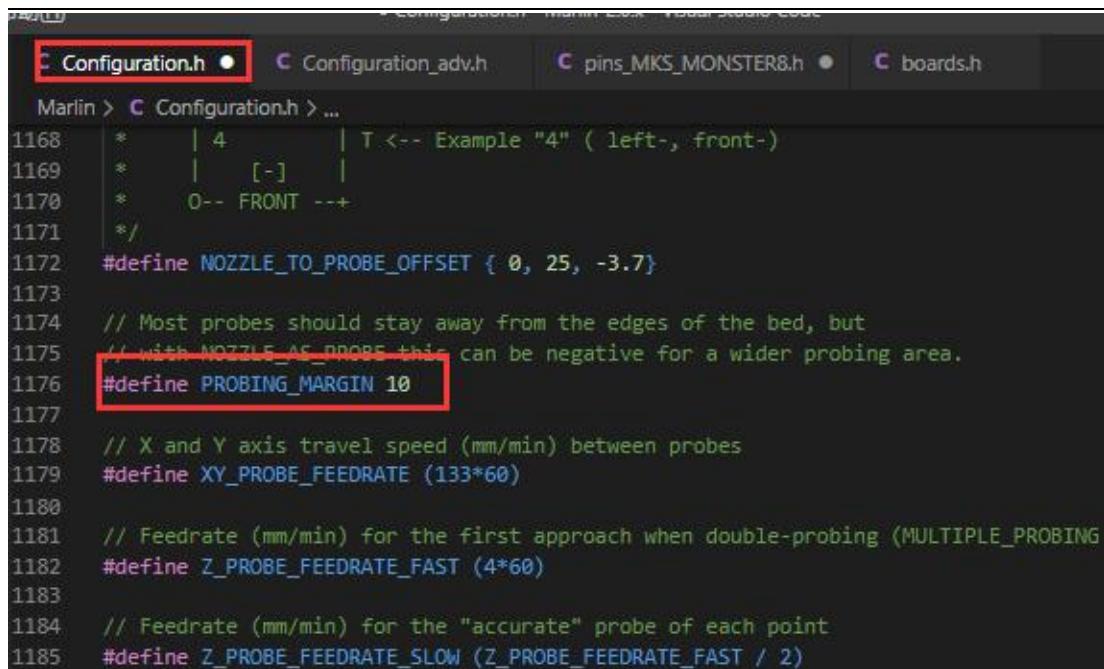
Based on Voron2.4, the X and Y axis offsets are 0 and 25, and the Z offset (z_offset) can be tested and adjusted after leveling.

```
Configuration.h • Configuration_adv.h pins_MKS_MONSTER8.h boards.h

Marlin > Configuration.h ...
1165 *      | 2      | +-- Example "2" (left-, back+)
1166 *      F | [-] N [+]| G <- Nozzle
1167 *      T |      3  | H <- Example "3" (right+, front-)
1168 *      | 4      | T <- Example "4" (left-, front-)
1169 *      | [-]    |
1170 *      O-- FRONT ---+
1171 */
1172 #define NOZZLE_TO_PROBE_OFFSET { 0, 25, -3.7}
1173
1174 // Most probes should stay away from the edges of the bed, but
1175 // with NOZZLE_AS_PROBE this can be negative for a wider probing area.
1176 #define PROBING_MARGIN 10
1177
1178 // X and Y axis travel speed (mm/min) between probes
1179 #define XY_PROBE_FEEDRATE (133*60)
1180
1181 // Feedrate (mm/min) for the first approach when double-probing (MULTIPLE_P
```

8.5 Set the distance between the sensor and the edge of the printing platform

The default value is 10 (Note: This value cannot be set too small, too small will cause the sensor to exceed the range of the platform during leveling, resulting in leveling failure)



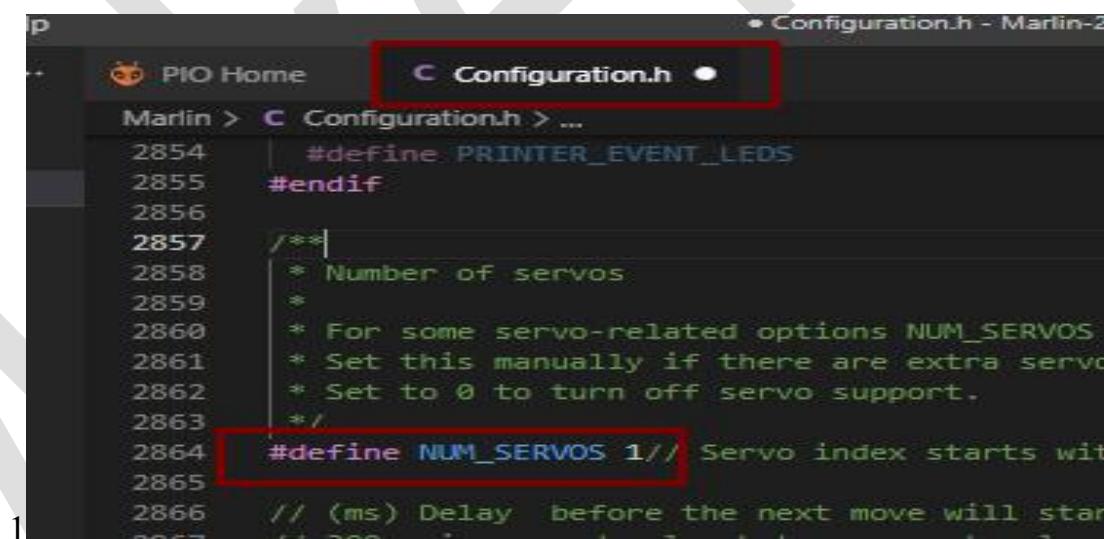
```

Marlin > C Configuration.h > ...
1168 * | 4 | T <-- Example "4" ( left-, front- )
1169 * | [-] |
1170 * 0-- FRONT --+
1171 */
1172 #define NOZZLE_TO_PROBE_OFFSET { 0, 25, -3.7 }
1173
1174 // Most probes should stay away from the edges of the bed, but
1175 // with NOZZLE_AS_PROBE this can be negative for a wider probing area.
1176 #define PROBING_MARGIN 10
1177
1178 // X and Y axis travel speed (mm/min) between probes
1179 #define XY_PROBE_FEEDRATE (133*60)
1180
1181 // Feedrate (mm/min) for the first approach when double-probing (MULTIPLE_PROBING)
1182 #define Z_PROBE_FEEDRATE_FAST (4*60)
1183
1184 // Feedrate (mm/min) for the "accurate" probe of each point
1185 #define Z_PROBE_FEEDRATE_SLOW (Z_PROBE_FEEDRATE_FAST / 2)

```

8.6 Enable the servos

Enable the servos in the configuration file



```

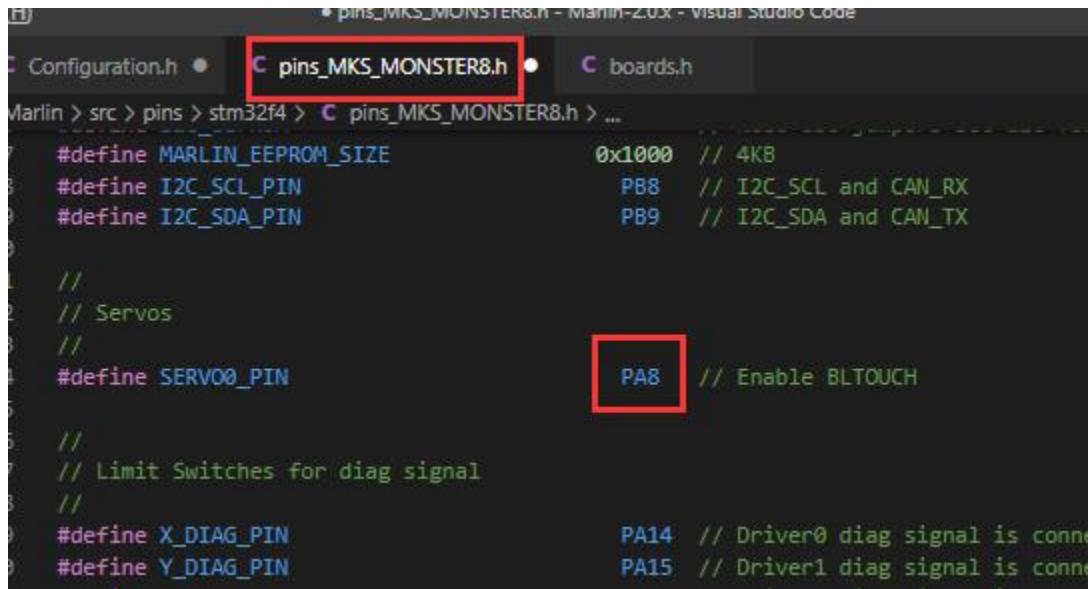
Marlin > C Configuration.h > ...
2854 #define PRINTER_EVENT_LEDS
2855 #endif
2856
2857 /**
2858 * Number of servos
2859 *
2860 * For some servo-related options NUM_SERVOS
2861 * Set this manually if there are extra servos
2862 * Set to 0 to turn off servo support.
2863 */
2864 #define NUM_SERVOS 1 // Servo index starts with 1
2865
2866 // (ms) Delay before the next move will start
2867 // 200 ms is a good value for most printers

```

8.7 Set servo pin port

Add the servos pin in pins_MONSTER8.h file (the default configuration

is PA8)



```
#define MARLIN_EEPROM_SIZE          0x1000 // 4KB
#define I2C_SCL_PIN                  PB8   // I2C_SCL and CAN_RX
#define I2C_SDA_PIN                  PB9   // I2C_SDA and CAN_TX

// Servos
// ...
#define SERVO0_PIN                   PA8   // Enable BLTOUCH

// Limit Switches for diag signal
// ...
#define X_DIAG_PIN                   PA14  // Driver0 diag signal is connected to pin PA14
#define Y_DIAG_PIN                   PA15  // Driver1 diag signal is connected to pin PA15
```

8.8 Enable 4Z automatic leveling

Enable 4Z automatic leveling in the advanced configuration file

```

Configuration_adv.h X pins_MKS_MONSTER8.h boards.h
Marlin > Configuration_adv.h > ...
850  * Add the G34 command to align multiple Z steppers using a bed probe.
851  */
852 #define Z_STEPPER_AUTO_ALIGN
853 #if ENABLED(Z_STEPPER_AUTO_ALIGN)
854  // Define probe X and Y positions for Z1, Z2 [, Z3 [, Z4]]
855  // If not defined, probe limits will be used.
856  // Override with 'M422 S<index> X<pos> Y<pos>'
857  // #define Z_STEPPER_ALIGN_XY { { 10, 190 }, { 100, 10 }, { 190, 190 } }
858
859 /**
860  * Orientation for the automatically-calculated probe positions.
861  * Override Z stepper align points with 'M422 S<index> X<pos> Y<pos>'
862  *
863  * 2 Steppers: (0)      (1)
864  *               |       |
865  *               1       2   |
866  *               |       |   |

```

8.9 Set the detection position of 4 motors

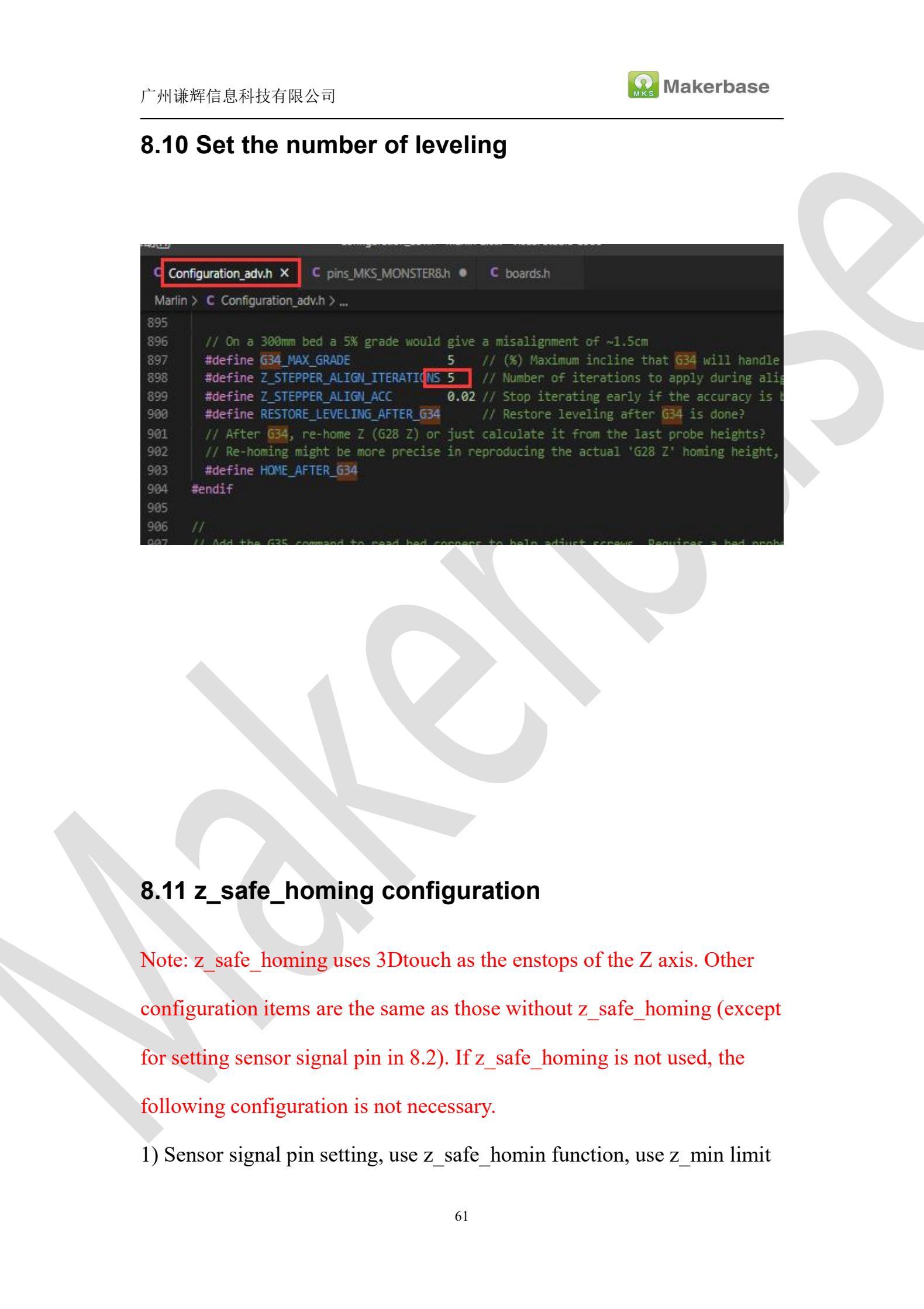
Set the detection location type to 0 in the advanced configuration file

```

Configuration_adv.h X pins_MKS_MONSTER8.h boards.h
Marlin > Configuration_adv.h > ...
869  *          | 3 | 1 | 2 | 1 | 2 |
870  *          |   |   |   |   |   |
871  *          | 1 | 2 | 2 | 3 | 1 |
872  *
873  * 4 Steppers: (0)      (1)      (2)      (3)
874  *               | 4 | 3 | 1 | 4 | 2 | 1 | 3 | 2 |
875  *               |   |   |   |   |   |   |   |   |
876  *               | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 1 |
877  */
878 #ifndef Z_STEPPER_ALIGN_XY
879 #define Z_STEPPERS_ORIENTATION 0
880#endif
881
882 // Provide Z stepper positions for more rapid convergence in bed alignment
883 // Requires triple stepper drivers (i.e., set NUM_Z_STEPPER_DRIVERS to 3)
884 // #define Z_STEPPER_ALIGN_KNOWN_STEPPER_POSITIONS
885 #if ENABLED(Z_STEPPER_ALIGN_KNOWN_STEPPER_POSITIONS)
886  // Define Stepper XY positions for Z1, Z2, Z3 corresponding to

```

8.10 Set the number of leveling



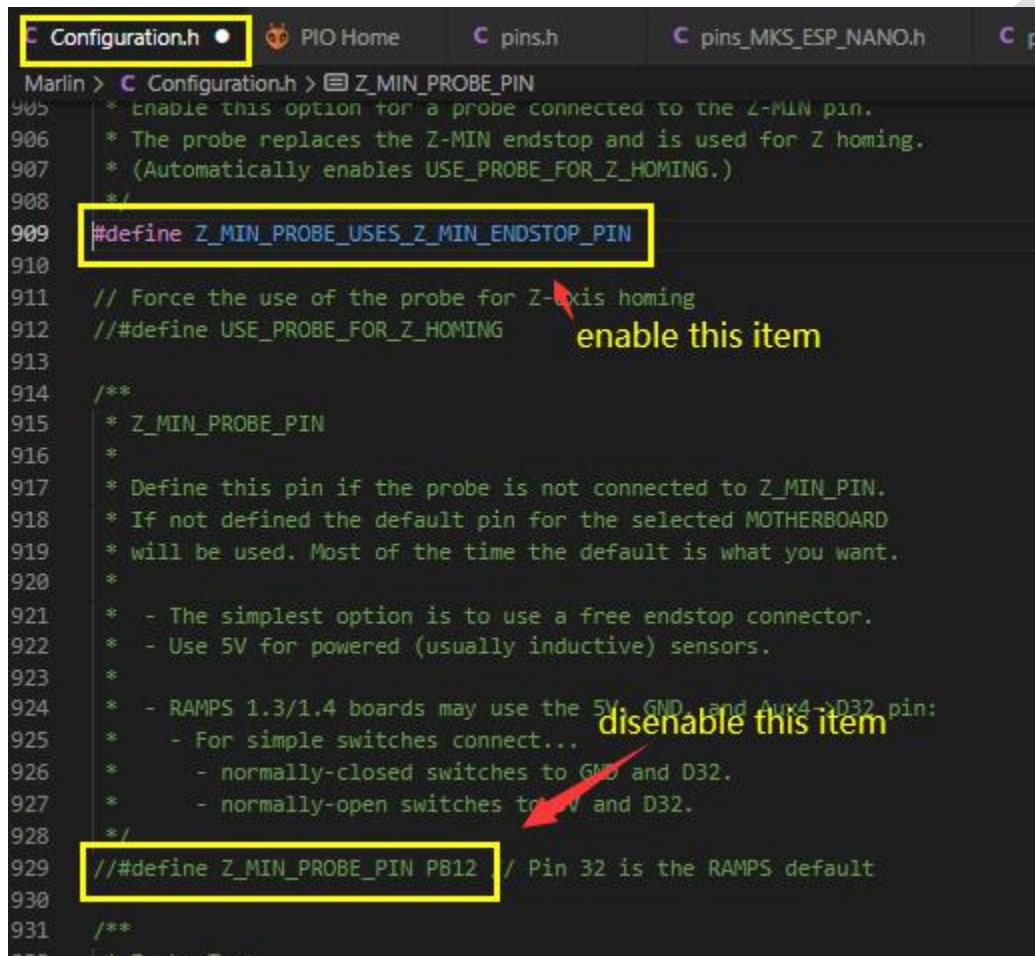
```
Configuration_adv.h X pins_MKS_MONSTER8.h • boards.h
Marlin > Configuration_adv.h ...
895
896 // On a 300mm bed a 5% grade would give a misalignment of ~1.5cm
897 #define G34_MAX_GRADE      5      // (%) Maximum incline that G34 will handle
898 #define Z_STEPPER_ALIGN_ITERATIONS 5 // Number of iterations to apply during align
899 #define Z_STEPPER_ALIGN_ACC    0.02 // Stop iterating early if the accuracy is b
900 #define RESTORE_LEVELING_AFTER_G34 // Restore leveling after G34 is done?
901 // After G34, re-home Z (G28 Z) or just calculate it from the last probe heights?
902 // Re-homing might be more precise in reproducing the actual 'G28 Z' homing height,
903 #define HOME_AFTER_G34
904 #endif
905
906 /**
907 // Add the G35 command to read bed sensors to help adjust screws. Requires a bed probe.
```

8.11 z_safe_homing configuration

Note: z_safe_homing uses 3Dtouch as the enstops of the Z axis. Other configuration items are the same as those without z_safe_homing (except for setting sensor signal pin in 8.2). If z_safe_homing is not used, the following configuration is not necessary.

- 1) Sensor signal pin setting, use z_safe_homin function, use z_min limit

pin, you need to make sure that the configuration file is enabled #define
Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN



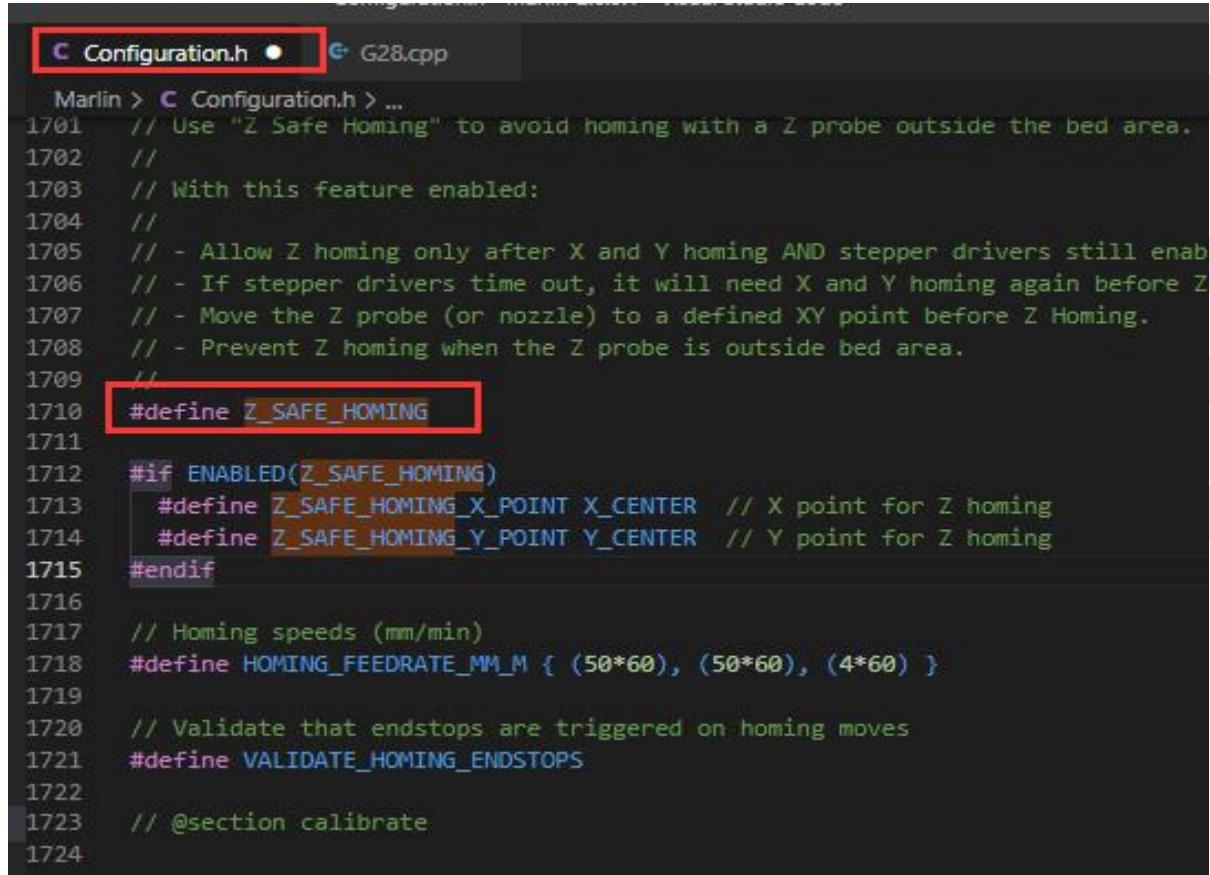
The screenshot shows a code editor with the file 'Configuration.h' open. The tab bar at the top includes 'Configuration.h' (which is the active tab), 'PIO Home', 'pins.h', 'pins_MKS_ESP_NANO.h', and 'pins_MKS_RAMP.h'. The code itself is a comment block for the Z_MIN_PROBE_PIN option. Several lines of code are highlighted with yellow boxes:

- Line 909: `#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN`
- Line 929: `//#define Z_MIN_PROBE_PIN PB12 / Pin 32 is the RAMPS default`

Annotations on the right side of the code:

- A red arrow points to the word 'enable' in the line `enable this item`.
- A red arrow points to the word 'disnable' in the line `disnable this item`.

2) enable z_safe_homing in the configuration file



```
Marlin > C Configuration.h > ...
1701 // Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
1702 //
1703 // With this feature enabled:
1704 //
1705 // - Allow Z homing only after X and Y homing AND stepper drivers still enable
1706 // - If stepper drivers time out, it will need X and Y homing again before Z homing.
1707 // - Move the Z probe (or nozzle) to a defined XY point before Z Homing.
1708 // - Prevent Z homing when the Z probe is outside bed area.
1709 //
1710 #define Z_SAFE_HOMING
1711
1712 #if ENABLED(Z_SAFE_HOMING)
1713     #define Z_SAFE_HOMING_X_POINT X_CENTER // X point for Z homing
1714     #define Z_SAFE_HOMING_Y_POINT Y_CENTER // Y point for Z homing
1715 #endif
1716
1717 // Homing speeds (mm/min)
1718 #define HOMING_FEEDRATE_MM_M { (50*60), (50*60), (4*60) }
1719
1720 // Validate that endstops are triggered on homing moves
1721 #define VALIDATE_HOMING_ENDSTOPS
1722
1723 // @section calibrate
1724
```

9. automatic leveling operation on LCD screen

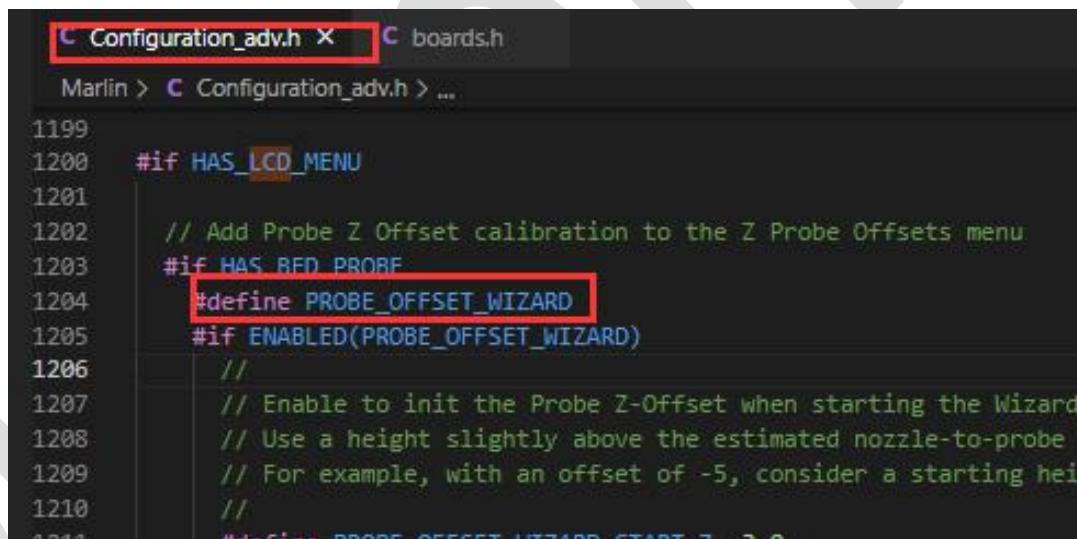
Press Rotary Encoder→Motion→Auto Z Alignment on the screen to start automatic 4Z automatic leveling (the Z-axis motor will automatically adjust to a plane relative to the platform during the leveling process. The default firmware performs 5 levels of leveling. Please try to install it as level as possible, otherwise it may cause the Z-axis to tilt greatly after automatic leveling, and the Z-axis will slide down after the

power is turned off or after the motor is unlocked. In this case, you need to level each time before printing)

10. z_offset adjustment

10.1 z_offset adjustment wizard configuration

Enable #define PROBE_OFFSET_WIZARD in the advanced configuration file (enabled by default firmware)



```
Configuration_adv.h X boards.h
Marlin > Configuration_adv.h > ...
1199
1200  #if HAS_LCD_MENU
1201
1202      // Add Probe Z Offset calibration to the Z Probe Offsets menu
1203  #if HAS_BED_PROBE
1204      #define PROBE_OFFSET_WIZARD
1205      #if ENABLED(PROBE_OFFSET_WIZARD)
1206          //
1207          // Enable to init the Probe Z-Offset when starting the Wizard
1208          // Use a height slightly above the estimated nozzle-to-probe
1209          // For example, with an offset of -5, consider a starting hei
1210          //
1211          //          PROBE_OFFSET_WIZARD_START_Z 1.0
1212
```

10.2 Operate on the LCD screen to adjust z_offset

Perform automatic leveling before adjusting z_offset. After leveling,

press the rotary button on the main interface → enter configuration → advanced configuration → probe offset → Z Probe Wizard → then rotate the encoder and slowly lower the Z axis until the nozzle Keep the distance of A4 paper thickness from the platform → click Finish → then return to the configuration interface → save settings, load settings.

11. LCD shell, motherboard mounting bracket stl file download link

MKS MINI12864 V3.0 shell installation download link:

<https://www.thingiverse.com/thing:4918948>

Download link of motherboard mounting bracket:

<https://www.thingiverse.com/thing:4977292>

12. FAQ

1. After updating the firmware, how to deal if the parameters such as

pulse and maximum speed displayed on the LCD screen are incorrect?

Answer: Enter the advanced setting interface on the screen, initialize eeprom, and then return to the setting interface, save data, and load data

2. How to deal with the motherboard cannot be connected to the host computer of the computer?

Answer: Confirm whether the serial port of the motherboard in the configuration file is correct. MKS MONSTER8V1.0 uses serial port-1; After the motherboard is connected to the computer, enter the device manager to check whether the computer recognizes the com port of the motherboard. If so, you can restart the host computer. The baud rate set by the host computer needs to be consistent with the configuration file setting; if it cannot be recognized, check the usb Whether the connection is bad.

3. How to deal with the TMC error on the screen?

Answer: The TMC driver error message appears on the screen because the motherboard firmware has enabled the TMC driver uart mode, and the communication between the motherboard and the driver failed.

1). Confirm whether the motherboard uses the TMC driver (TMC2208, 2209, 2225, 2226), if it is not used, you need to shield the TMC driver in

the configuration file

2). The TMC driver is used and the firmware configuration is correct.

You need to confirm whether the hardware has been set to uart mode, that is, whether the jumper under the driver is correct, see 3.4 for details

3). If the above configuration is correct, there may be a driver problem, you can use the host computer to send the command M122 to check, it shows that the communication of ok is successful, and the communication of allow is failed.

