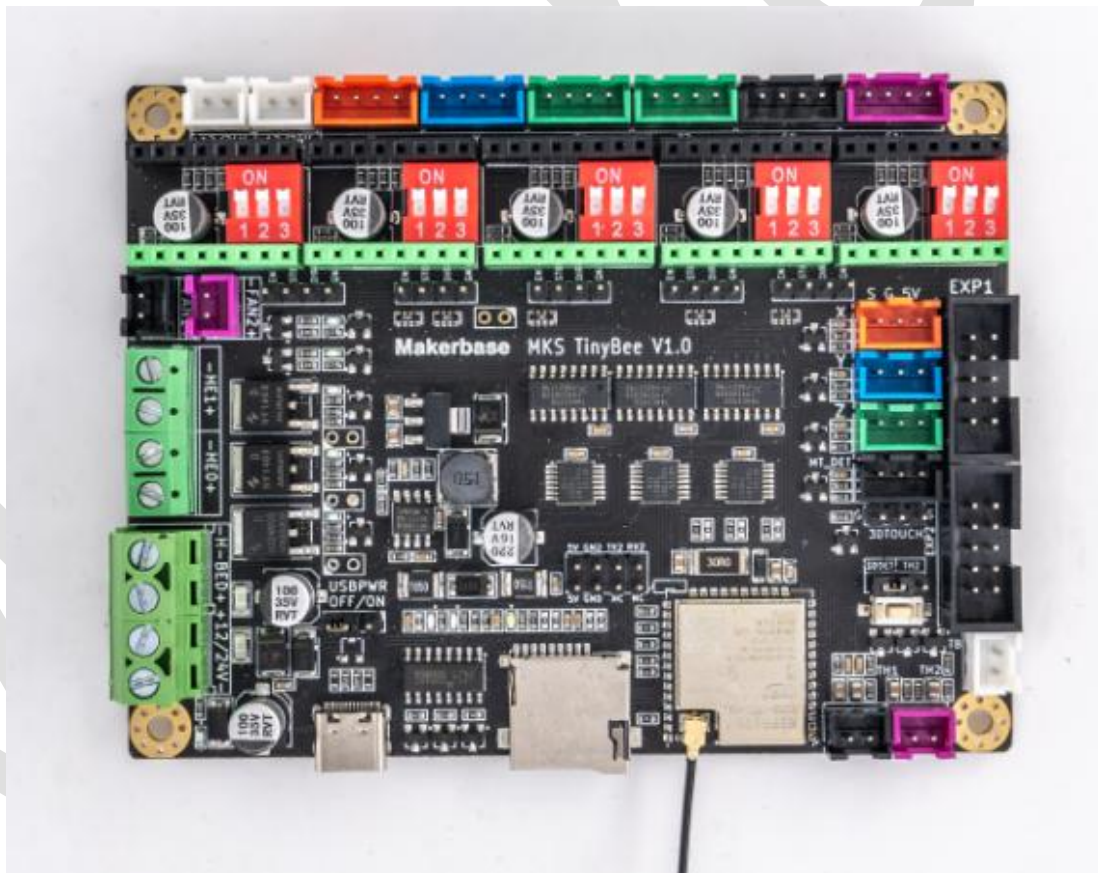




Makerbase

**Guangzhou Qianhui Information
Technology Co., Ltd.**

MKS TinyBee V1.0 datasheet



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-9fo&marketplaceID=ATVPDKIKX0DER

content

About us:.....	2
Facebook:	2
YouTub:.....	2
Github:.....	2
AliExpress:	2
Amazon:.....	2
1. Product Brief.....	5
1.1 Features and advantages.....	6
1.2 Motherboard parameters.....	8
1.3 Wiring diagram.....	9
1.4 Dimensions.....	9
2. Firmware download, compile and update.....	10
2.1 Firmware download.....	10
2.2 Firmware compilation.....	10
2.3 Firmware upload.....	11

- 3. Drive subdivision settings.....12
 - 3.1 A4988 driver subdivision settings..... 12
 - 3.2 TMC2208, TMC2209, TMC2226 common mode..... 13
 - Settings..... 13
 - 3.3 TMC2225 normal mode setting.....14
- 4.Marlin firmware configuration..... 15
 - 4.1 Basic firmware configuration (must configure items)..... 15
 - 4.1.1Motherboard configuration.....15
 - 4.1.2 Serial port configuration..... 16
 - 4.1.3 Screen configuration..... 18
 - 4.1.5 Enable eeprom..... 21
 - 4.1.6 Compile environment configuration..... 22
 - 4.2 Machine parameter configuration (set according to machine parameter)..... 22
 - 4.2.1 Number of extrusion heads configuration.....22
 - 4.2.2 Thermal type configuration..... 23
 - 4.2.3 Enstop level type configuration.....24
 - 4.2.4 Pulse setting..... 25
 - 4.2.5 Zero return direction setting..... 26
 - 4.2.6 Print platform range setting..... 26
 - 4.2.7 Motor direction setting..... 26
 - 4.3 Advanced configuration.....27

4.3.1 POWER_LOSS_RECOVERY.....	27
4.3.2 FILAMENT_RUNOUT_SENSOR.....	28
4.3.3 WIFI configuration.....	32
5. 3dtouch automatic leveling function.....	32
5.1 Set the sensor trigger level.....	32
5.2 Set sensor signal pin.....	33
5.3 Enable BLTOUCH.....	35
5.4 Set the offset between the probe and the extrusion.....	35
5.5 Set the distance between the sensor and the edge of the printing platform during leveling.....	36
5.6 Enable automatic leveling.....	37
5.7 Set the number of grid points for leveling.....	37
5.8 Enable the servo.....	38
5.9 Add auto-leveling data enable code.....	38
5.11 enable z_safe_homing.....	39
6.WEB connection and settings.....	40
7、FAQ.....	44

1. Product Brief

MKS TinyBee V1.0 motherboard is a 32-bit main control board launched

by the makerbase team to meet market needs. The motherboard supports wifi function without additional wifi module, supports web page control, supports LCD2004, LCD mini12864, MKS mini12864 V3. 0, LCD12864, support serial screen control

1.1 Features and advantages

1. Support wifi control and wireless file transfer;
2. The driver subdivision setting is the dial code setting, which is more convenient for the driver subdivision setting;
3. The main board PCB adopts immersion gold technology, which has better stability;
3. The user can replace the motor drive by himself, supporting 4988, 8825, 8729, TMC2208, TMC2209, TMC2225, TMC2226;
4. Reserve an external drive signal, and an external large drive can be used to drive 57 and 86 motors;
5. Using high-quality MOSFET tubes, the heat dissipation effect is better, and the long-term work is stable;
6. Use a dedicated power chip, support 12V-24V power input, and solve the problem of heating and insufficient power supply of the Ramps voltage conversion chip;
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. Adopting CH340 serial port chip, under the premise of ensuring stability and reliability, reducing costs, and also solving the problem of difficult driver installation in the past 16U2;

9. Use the open source firmware Marlin2.0.X firmware;

10. Support LCD2004, LCD12864, MKS MINI12864 V1.0, MKS MINI12864 V3.0, support TFT24, TFT28, TFT32, TFT35, H43 touch screens developed by makers;

11. The XYZ axis uses different color terminals to correspond to the motor and limit switch, which is convenient for wiring;

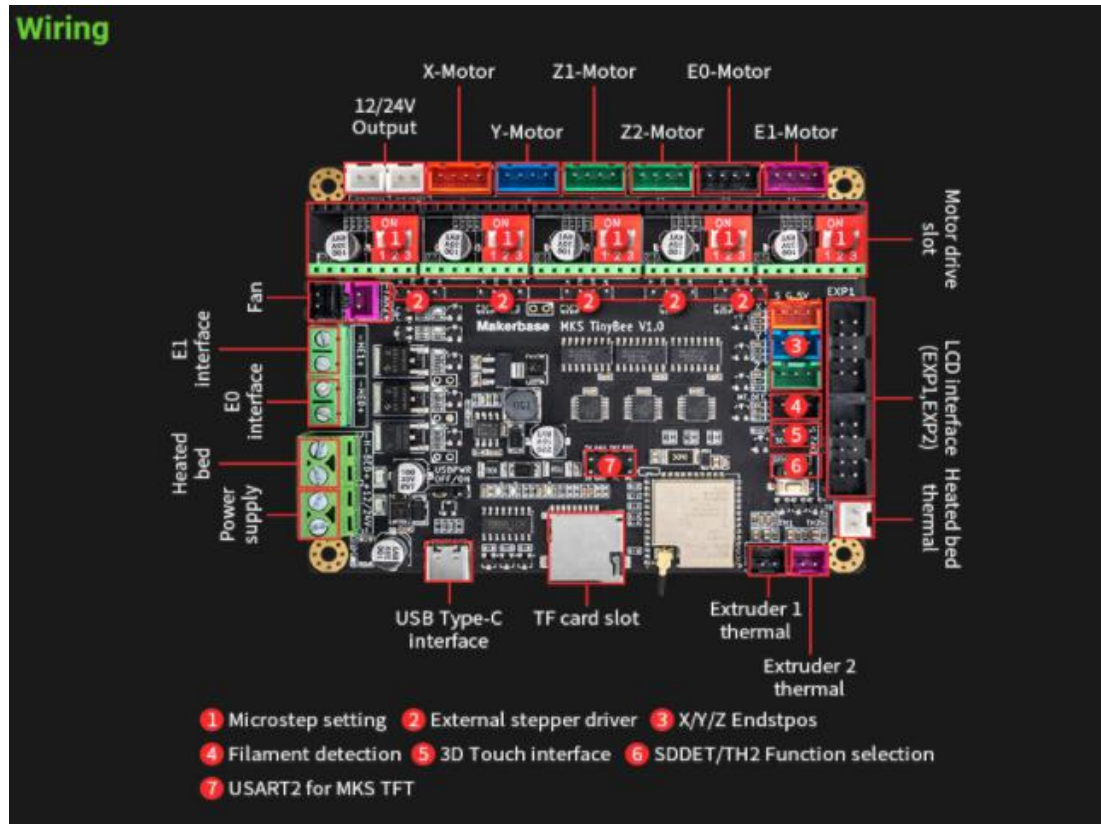
12. Support 3dtouch;

13. Onboard TF card holder, offline printing is more stable;

1.2 Motherboard parameters

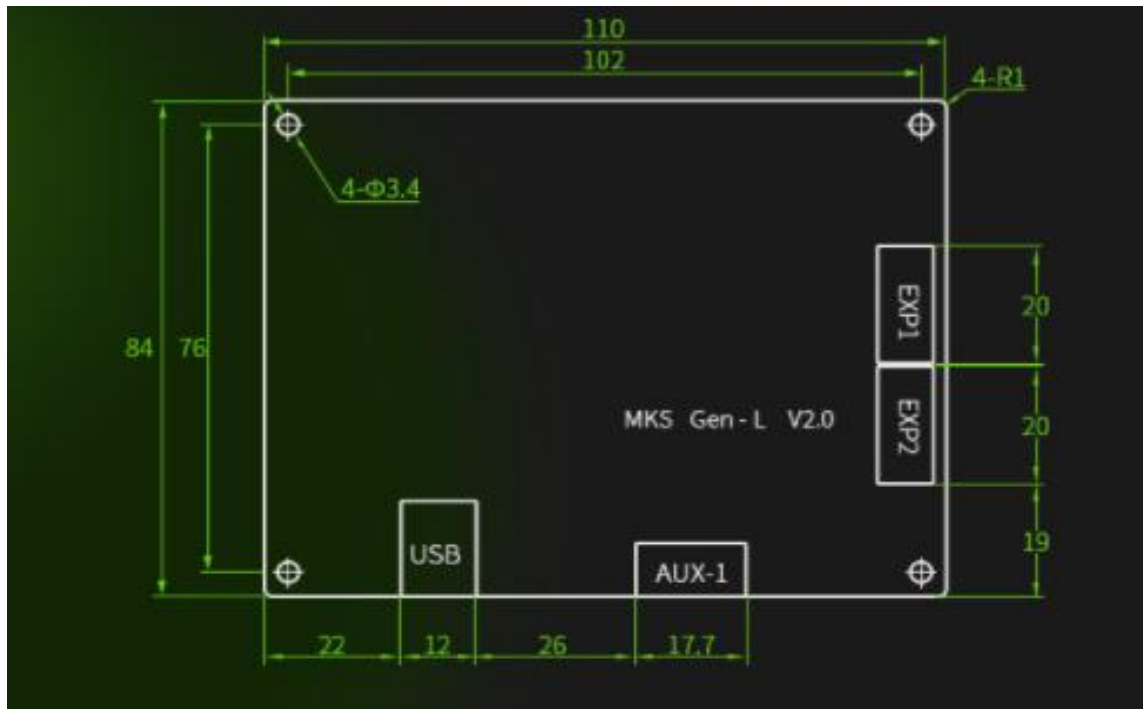
Motherboard type:	MKS TinyBee V1.0	microprocessor:	ESP32-WROOM-32U
physical dimension:	110*84	Mounting hole size:	102*76
Input voltage:	12V~24V 5A~15A	motor driver:	2208,2209,2225,2226,49 88,8825,8729
Temperature sensor interface:	NTC 100K	Support LCD/touch screen:	LCD2004、LCD12864、 MKS MINI12864 V1.0、 MKS MINI12864 V3.0、 MKS 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:	Update via USB connection to computer

1.3 Wiring diagram



1.4 Dimensions

Motherboard sizechart:



2. Firmware download, compile and update

2.1 Firmware download

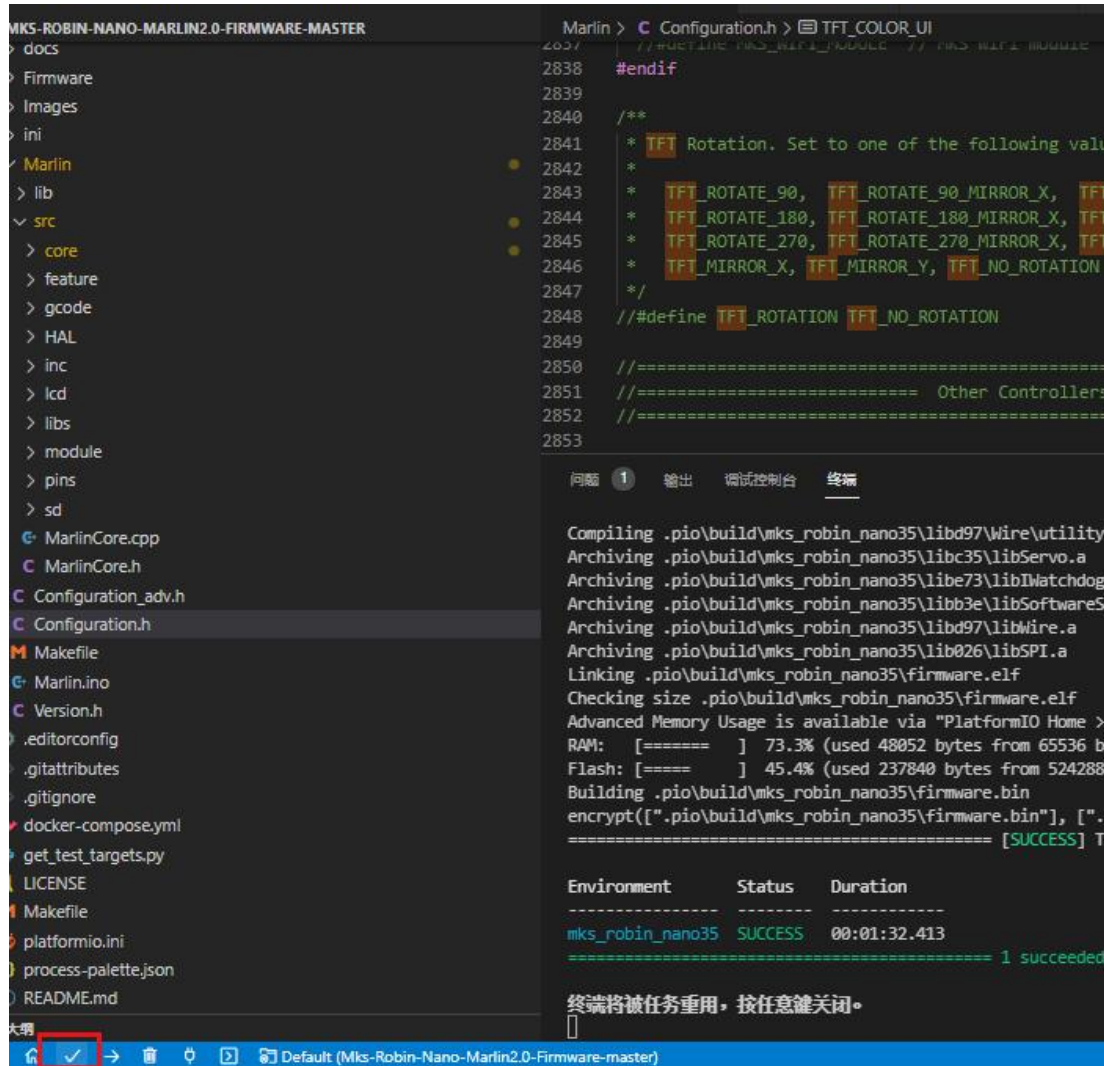
MKS TinyBee V1.0 firmware download link:

<https://github.com/makerbase-mks/MKS-TinyBee/blob/main/firmware/mks%20tinybee%20marlin.rar>

2.2 Firmware compilation

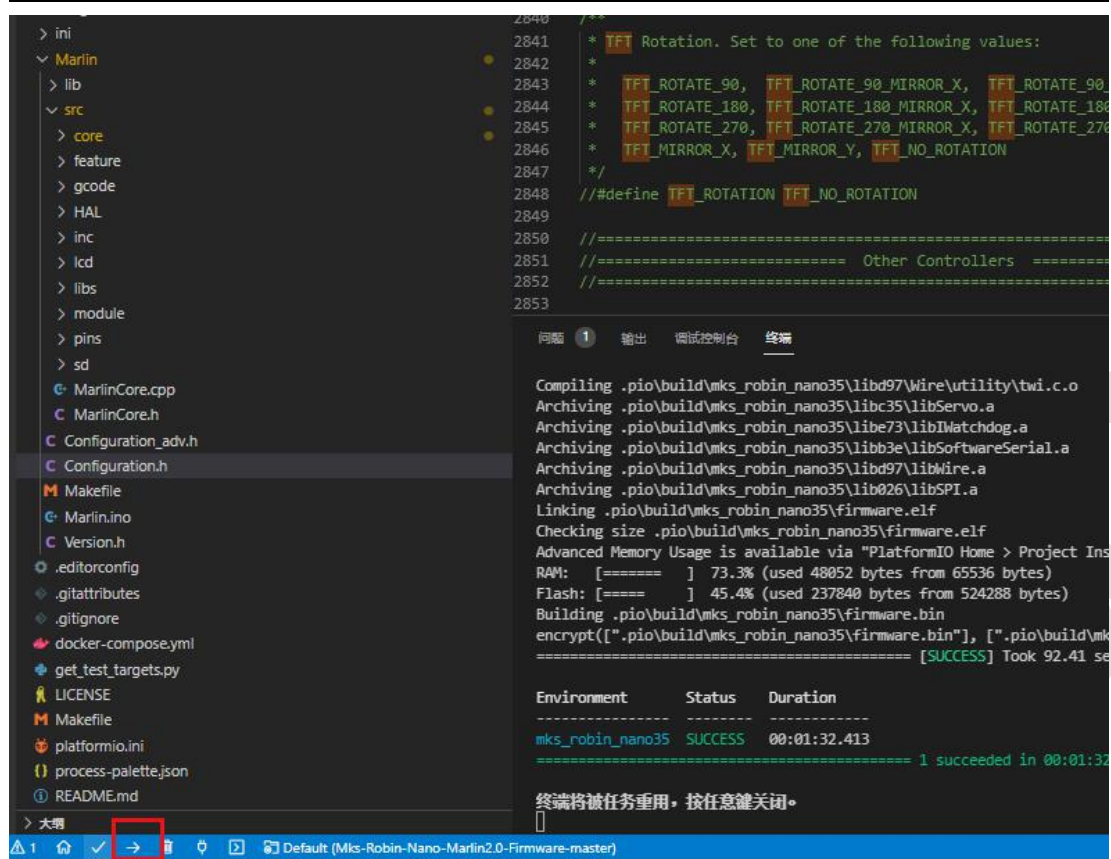
Firmware compilation:

After the firmware configuration is complete, click "✓" in the lower left corner of the VSCode page to start compiling.



2.3 Firmware upload

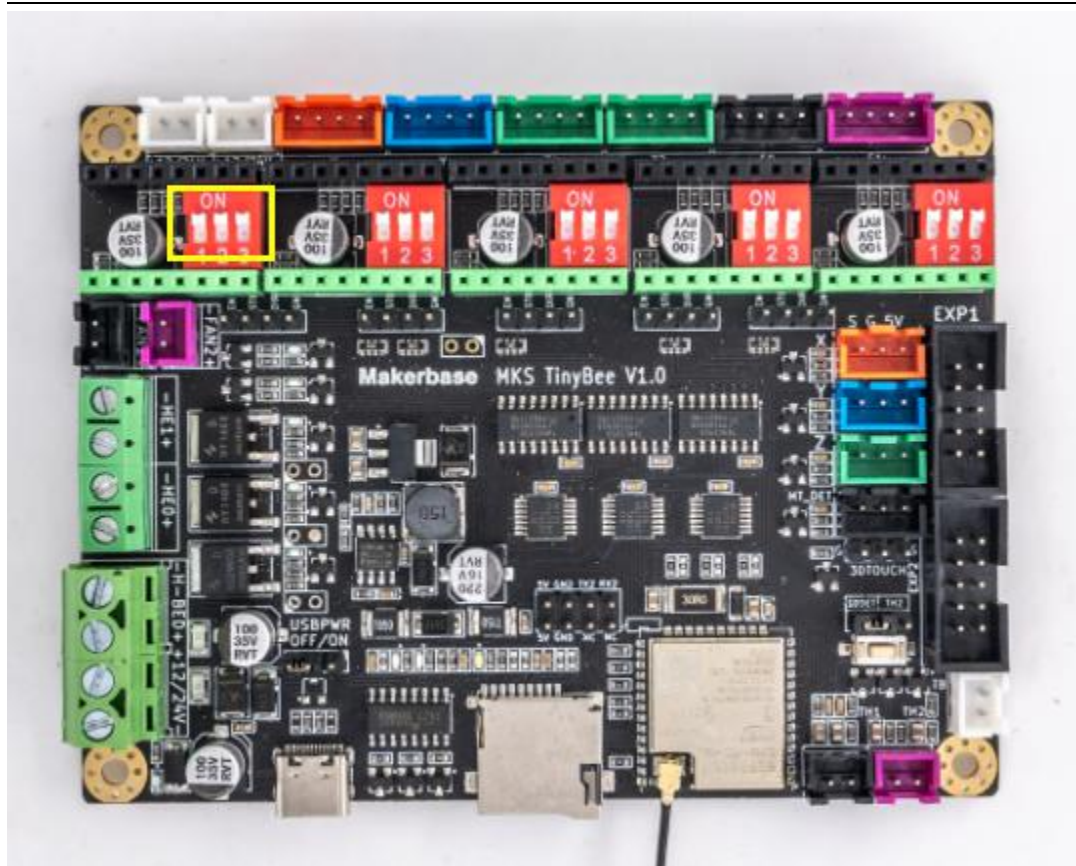
After the firmware is compiled, click "→" in the lower left corner of the VSCode page to start uploading the firmware.



3. Drive subdivision settings

3.1 A4988 driver subdivision settings

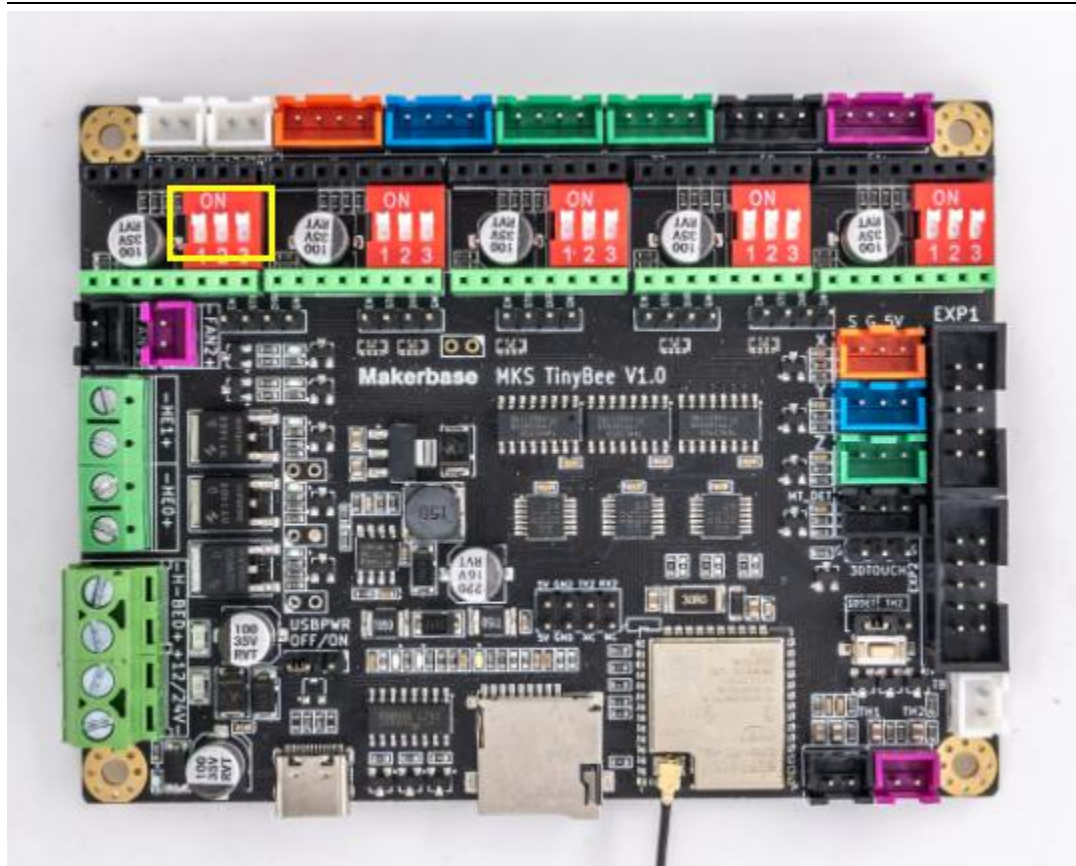
A4988 drives the subdivision setting, the three dials below the drive are dialed up to 16 subdivisions, as shown in the figure below (X-axis as an example):



3.2 TMC2208, TMC2209, TMC2226 common mode

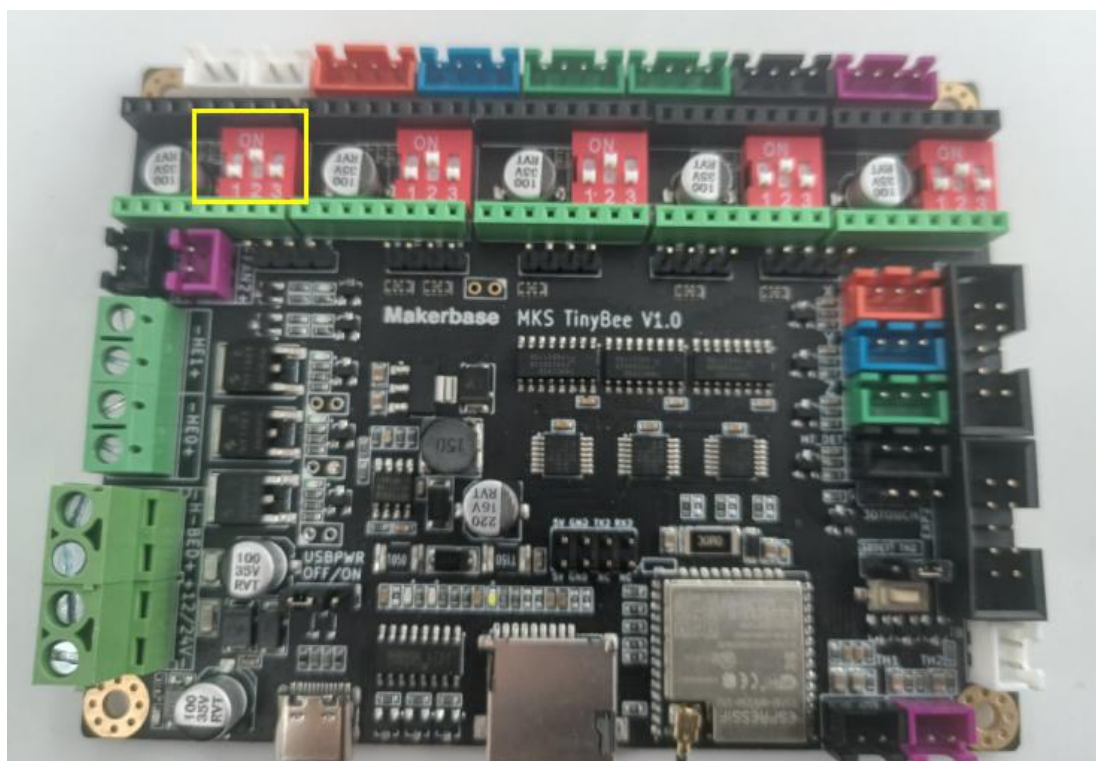
Settings

TMC2208, TMC2209, TMC2226 drive the three dial codes below to dial up to 16 subdivisions, as shown in the following figure (X-axis as an example):



3.3 TMC2225 normal mode setting

TMC2225 drives the subdivision setting. Drive the second dial below to dial up, and the first and third dials down are 16 subdivisions, as shown in the figure below (X-axis as an example):



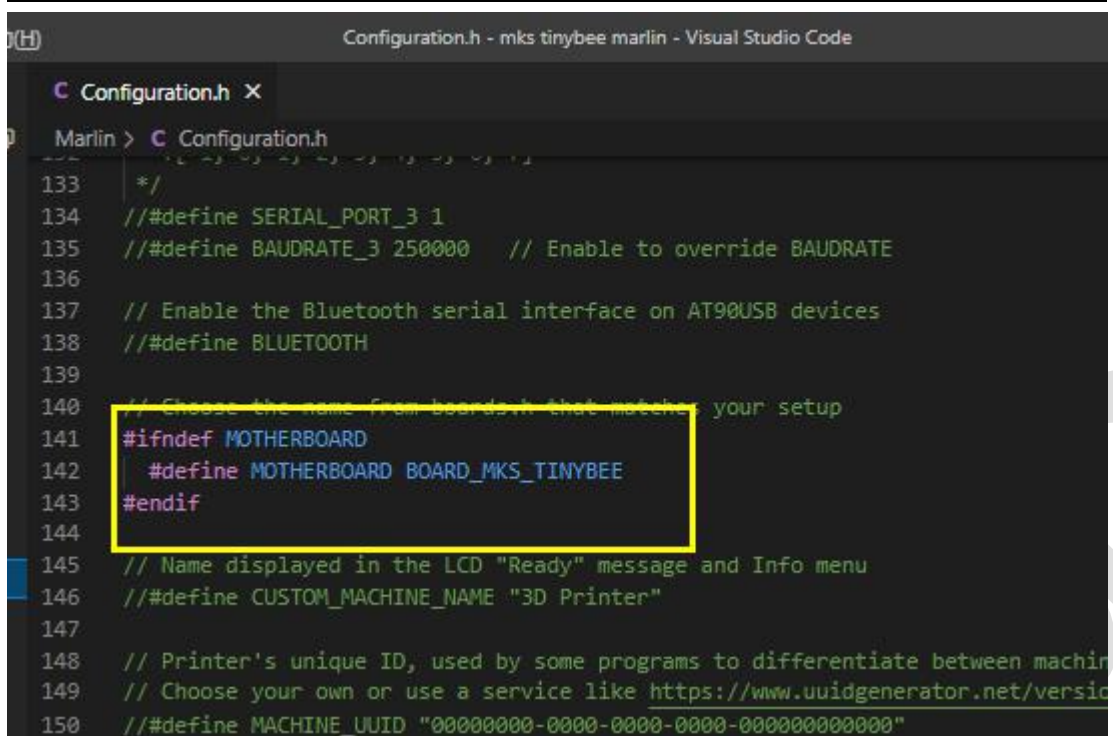
4. Marlin firmware configuration

4.1 Basic firmware configuration (must configure items)

4.1.1 Motherboard configuration

Configure the board in the configuration.h file as

`BOARD_MKS_TINYBEE`



```
Configuration.h - mks tinybee marlin - Visual Studio Code
C Configuration.h X
Marlin > C Configuration.h
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 that matches your setup
141  #ifndef MOTHERBOARD
142    #define MOTHERBOARD BOARD_MKS_TINYBEE
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"
```

4.1.2 Serial port configuration

Set the first serial port to 0 in the configuration.h file, the configuration error will cause the motherboard to be unable to connect to the computer; the second serial port is configured to -1, the web page will not be able to obtain the motherboard temperature information if the configuration error is incorrect; then baud The rates are all set to 115200.


```

C Configuration_adv.h  C Configuration.h  C boards.h  platformio
Marlin > C Configuration.h > SERIAL_PORT
95 // @section machine
96
97 /**
98  * Select the serial port on the board to use for communication
99  * This allows the connection of wireless adapters (for instance
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
102  *
103  * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
104  */
105 #define SERIAL_PORT 0
106
107 /**
108  * Serial Port Baud Rate
109  * This is the default communication speed for all serial ports
110  * Set the baud rate defaults for additional serial ports below
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  *
116  * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
117  */

```

```

Configuration.h - Mks-mifeng-Firmware - Visual Studio Code
开始  C Configuration.h M X  PIO Home  u8g_esp32_spi.cpp U  MarlinC
Marlin > C Configuration.h > ...
104 */
105 #define SERIAL_PORT 0
106
107 /**
108  * Serial Port Baud Rate
109  * This is the default communication speed for all serial ports.
110  * Set the baud rate defaults for additional serial ports below.
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  *
116  * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
117  */
118 #define BAUDRATE 115200
119 //#define BAUD_RATE_GCODE // Enable G-code M575 to set the baud rate
120
121 /**
122  * Select a secondary serial port on the board to use for communication with the
123  * Currently Ethernet (-2) is only supported on Teensy 4.1 boards.
124  * :[-2, -1, 0, 1, 2, 3, 4, 5, 6, 7]
125  */
126 #define SERIAL_PORT_2 -1
127 #define BAUDRATE_2 115200 // Enable to override BAUDRATE
128
129 /**
130  * Select a third serial port on the board to use for communication with the host
131  * Currently only supported for AVR, DUE, LPC1768/9 and STM32/STM32F1

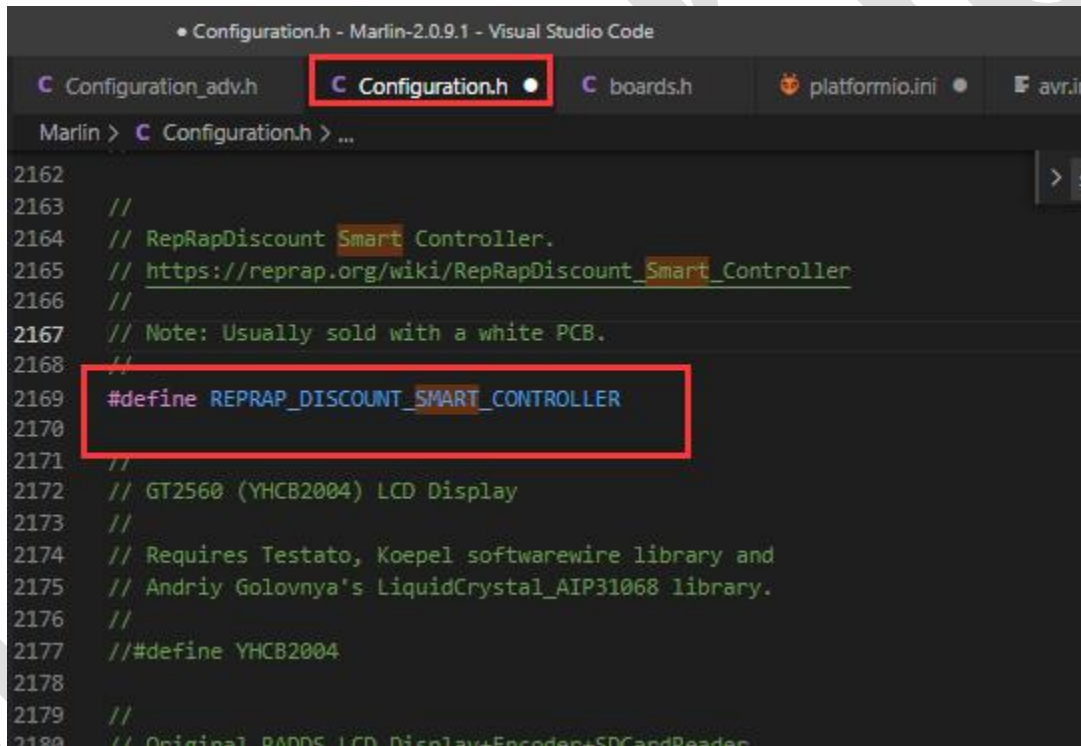
```

4.1.3 Screen configuration

(Note: LCD screen can only enable one of the following screens, enabling multiple at the same time will cause compilation error)

1. LCD 2004 configuration, enable in the configuration file

```
#define REPRAP_DISCOUNT_SMART_CONTROLLER
```

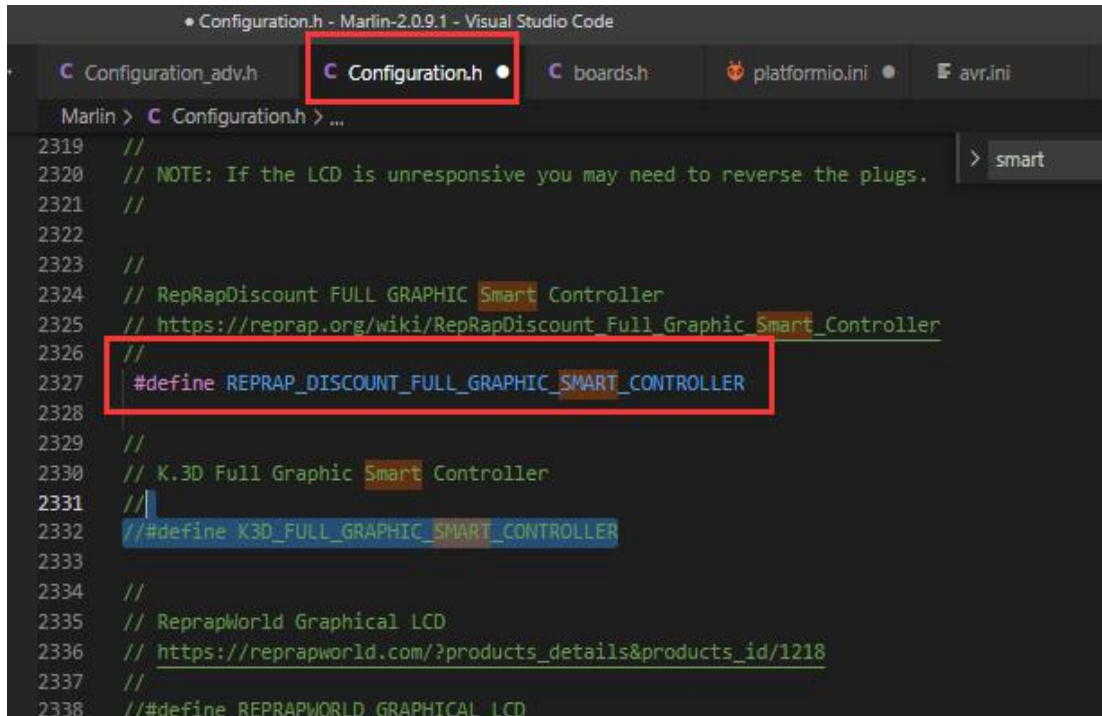


```
• Configuration.h - Marlin-2.0.9.1 - Visual Studio Code
C Configuration_adv.h C Configuration.h C boards.h platformio.ini avr.in
Marlin > C Configuration.h > ...
2162
2163 //
2164 // RepRapDiscount Smart Controller.
2165 // https://reprap.org/wiki/RepRapDiscount_Smart_Controller
2166 //
2167 // Note: Usually sold with a white PCB.
2168 //
2169 #define REPRAP_DISCOUNT_SMART_CONTROLLER
2170
2171 //
2172 // GT2560 (YHCB2004) LCD Display
2173 //
2174 // Requires Testato, Koepel softwarewire library and
2175 // Andriy Golovnya's LiquidCrystal_AIP31068 library.
2176 //
2177 // #define YHCB2004
2178
2179 //
2180 // Original RADD5 LCD Display+Encoder+SDCardReader
```

2. LCD12864 screen configuration, enabled in the configuration file

```
#define
```

```
REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
```

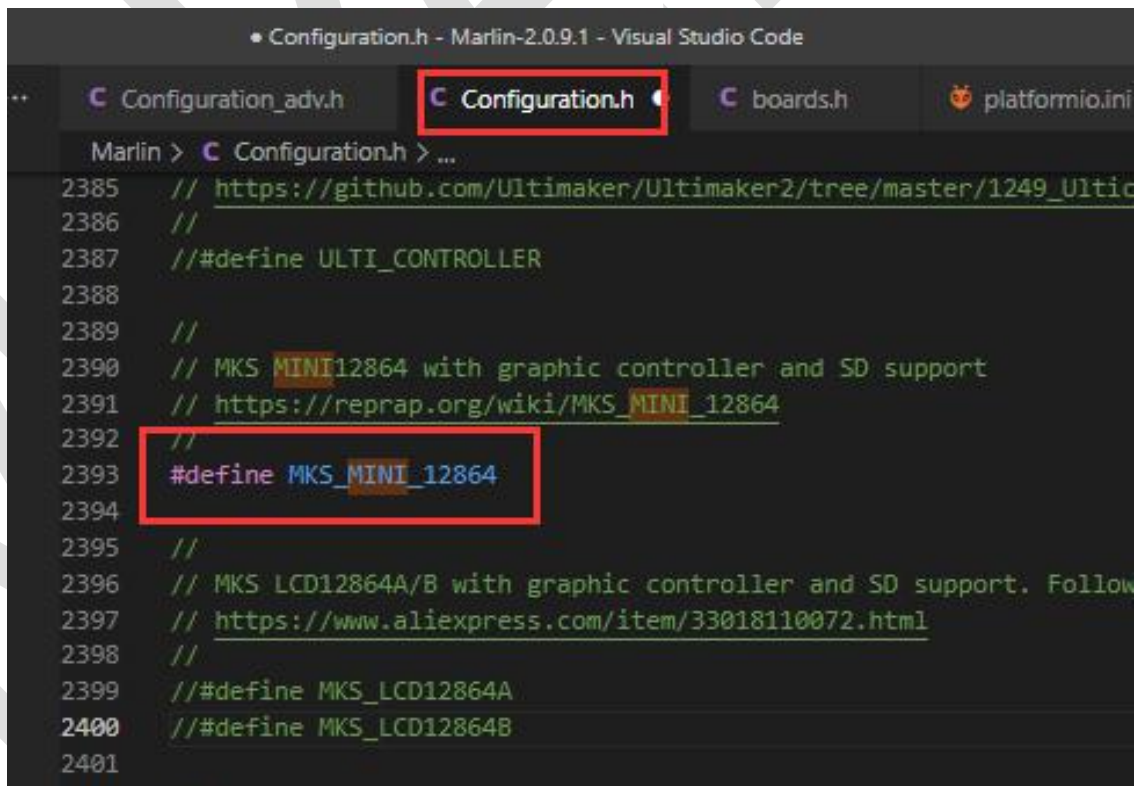


```

2319 //
2320 // NOTE: If the LCD is unresponsive you may need to reverse the plugs.
2321 //
2322 //
2323 //
2324 // RepRapDiscount FULL GRAPHIC Smart Controller
2325 // https://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller
2326 //
2327 #define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
2328 //
2329 //
2330 // K.3D Full Graphic Smart Controller
2331 //
2332 //#define K3D_FULL_GRAPHIC_SMART_CONTROLLER
2333 //
2334 //
2335 // ReprapWorld Graphical LCD
2336 // https://reprapworld.com/?products_details&products_id/1218
2337 //
2338 //#define REPRAPWORLD_GRAPHICAL_LCD
    
```

3. MKS MINI12864 V1.0 screen configuration

Enable in the configuration file #define MKS_MINI_12864

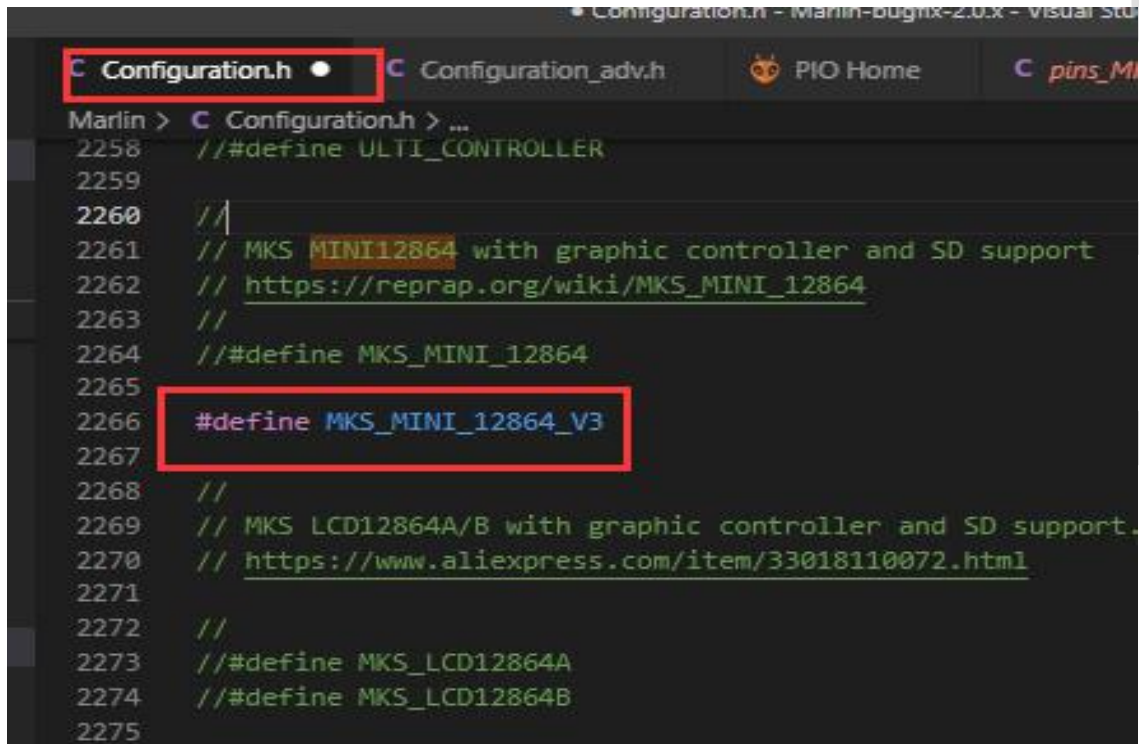


```

2385 // https://github.com/Ultimaker/Ultimaker2/tree/master/1249_Ulti
2386 //
2387 //#define ULTI_CONTROLLER
2388 //
2389 //
2390 // MKS MINI12864 with graphic controller and SD support
2391 // https://reprap.org/wiki/MKS_MINI_12864
2392 //
2393 #define MKS_MINI_12864
2394 //
2395 //
2396 // MKS LCD12864A/B with graphic controller and SD support. Follow
2397 // https://www.aliexpress.com/item/33018110072.html
2398 //
2399 //#define MKS_LCD12864A
2400 //#define MKS_LCD12864B
2401 //
    
```

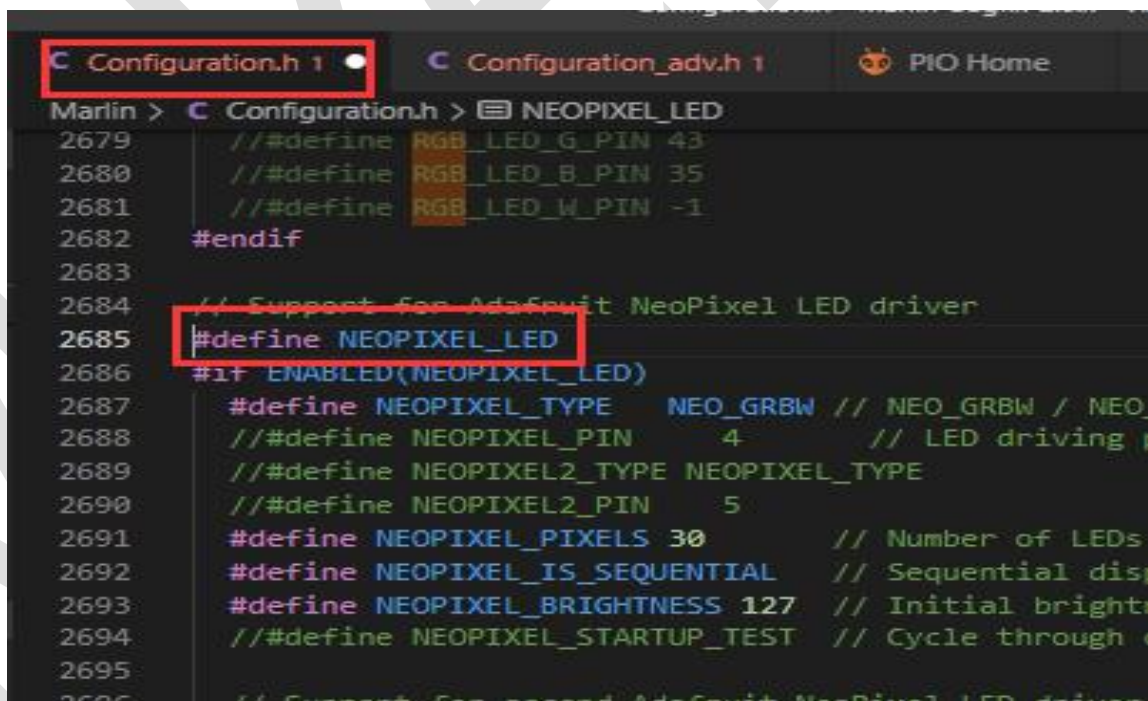

4. MKS MINI12864 V3.0 screen configuration

Enable in the configuration file MKS MINI12864 V3



```
Configuration.h - Marlin-bugfix-2.0.x - Visual Studio Code
C Configuration.h • C Configuration_adv.h PIO Home 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
```

Enable RGB in the configuration file



```
Configuration.h 1 Configuration_adv.h 1 PIO Home
Marlin > C Configuration.h > NEOPIXEL_LED
2679 // #define RGB_LED_G_PIN 43
2680 // #define RGB_LED_B_PIN 35
2681 // #define RGB_LED_W_PIN -1
2682 #endif
2683
2684 // Support for Adafruit NeoPixel LED driver
2685 #define NEOPIXEL_LED
2686 #if ENABLED(NEOPIXEL_LED)
2687 #define NEOPIXEL_TYPE NEO_GRBW // NEO_GRBW / NEO
2688 // #define NEOPIXEL_PIN 4 // LED driving
2689 // #define NEOPIXEL2_TYPE NEOPIXEL_TYPE
2690 // #define NEOPIXEL2_PIN 5
2691 #define NEOPIXEL_PIXELS 30 // Number of LEDs
2692 #define NEOPIXEL_IS_SEQUENTIAL // Sequential disp
2693 #define NEOPIXEL_BRIGHTNESS 127 // Initial brightn
2694 // #define NEOPIXEL_STARTUP_TEST // Cycle through
2695
2696 // Support for second Adafruit NeoPixel LED driver
```

4.1.4 Enable SD card

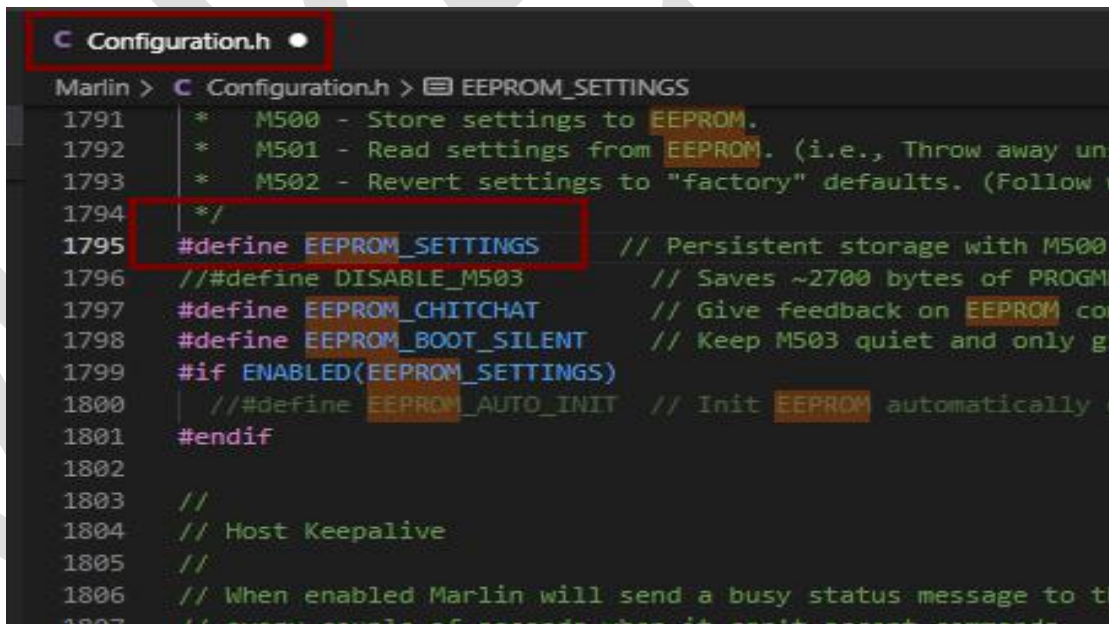
Enable #define SDSUPPORT in the configuration file



```
Configuration.h
Marlin > 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 /**
2071  * SD CARD: ENABLE CRC
2072  *
2073  * Use CRC checks and retries on the SD communication.
2074  */
2075 // #define SD_CHECK_AND_RETRY
2076
2077 /**
```

4.1.5 Enable eeprom

Enable in the configuration file #define EEPROM_SETTINGS

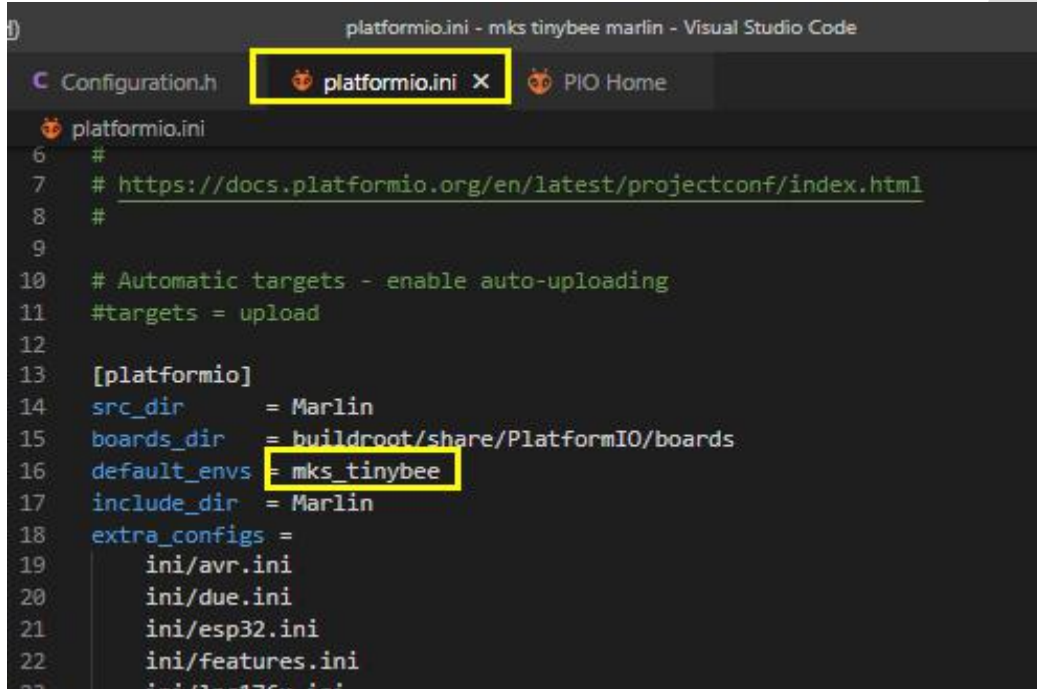


```
Configuration.h
Marlin > Configuration.h > EEPROM_SETTINGS
1791 * M500 - Store settings to EEPROM.
1792 * M501 - Read settings from EEPROM. (i.e., Throw away uns
1793 * M502 - Revert settings to "factory" defaults. (Follow w
1794 */
1795 #define EEPROM_SETTINGS // Persistent storage with M500
1796 // #define DISABLE_M503 // Saves ~2700 bytes of PROGM
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 c
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
```

4.1.6 Compile environment configuration

Configure the compilation environment in the platformio.ini file as

mks_tinybee



```
platformio.ini - mks tinybee marlin - Visual Studio Code
Configuration.h platformio.ini PIO Home
platformio.ini
6 #
7 # https://docs.platformio.org/en/latest/projectconf/index.html
8 #
9
10 # Automatic targets - enable auto-uploading
11 #targets = upload
12
13 [platformio]
14 src_dir = Marlin
15 boards_dir = buildroot/share/PlatformIO/boards
16 default_envs = mks_tinybee
17 include_dir = Marlin
18 extra_configs =
19     ini/avr.ini
20     ini/due.ini
21     ini/esp32.ini
22     ini/features.ini
23     ini/tec176x.ini
```

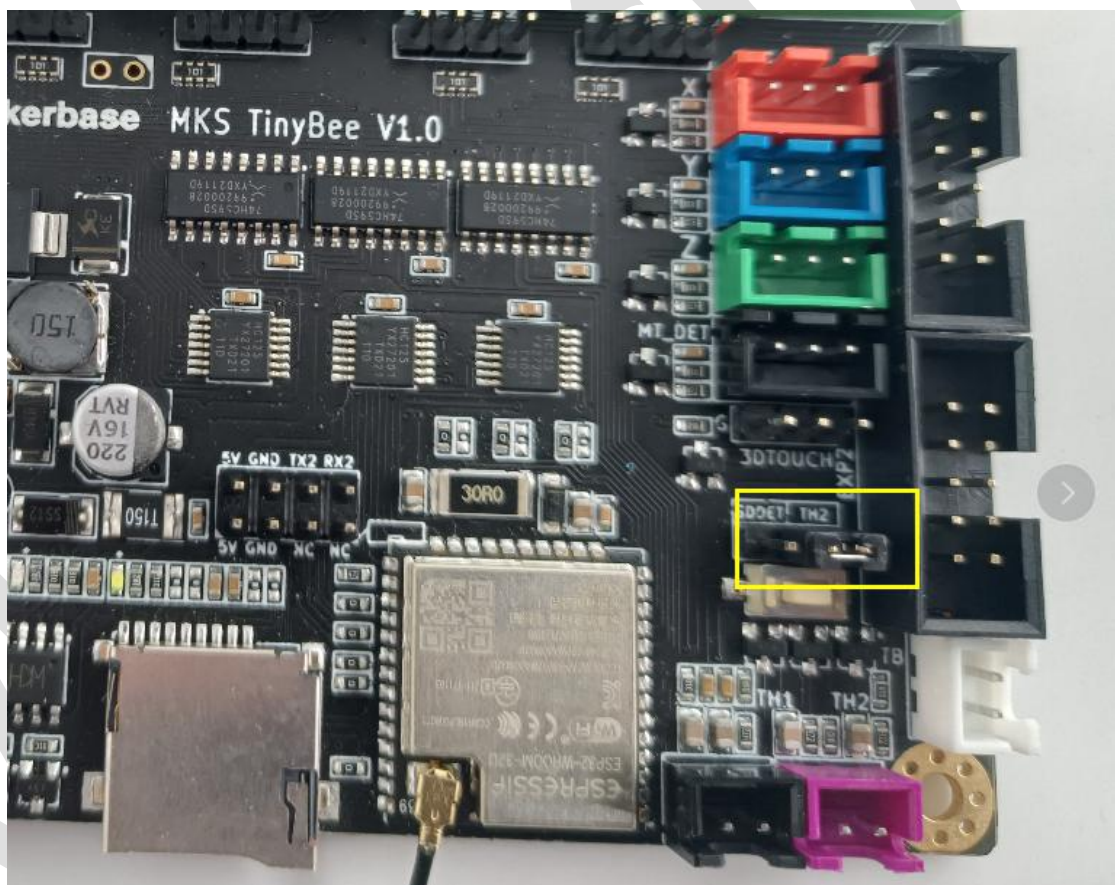
4.2 Machine parameter configuration (set according to machine parameter)

4.2.1 Number of extrusion heads configuration

MKS TinyBee V1.0 supports up to 2 extruders. When using dual extruders, you need to insert the extruder 2 thermal and SD card detection pin jumper on the motherboard to the right.


```

C Configuration.h X
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
    
```



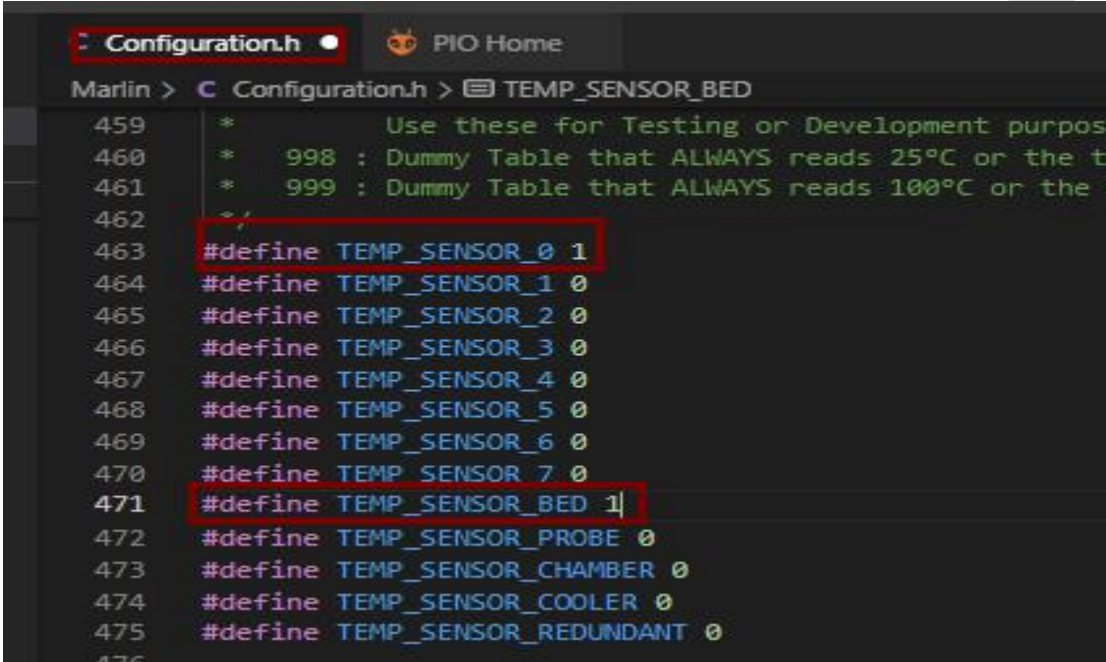
4.2.2 Thermal type configuration

MKS TinyBee V1.0 motherboard only supports 100K thermal sensitivity,

#define TEMP_SENSOR_0 is extrusion head 1, #define

TEMP_SENSOR_1 is extrusion head 2, #define TEMP_SENSOR_BED

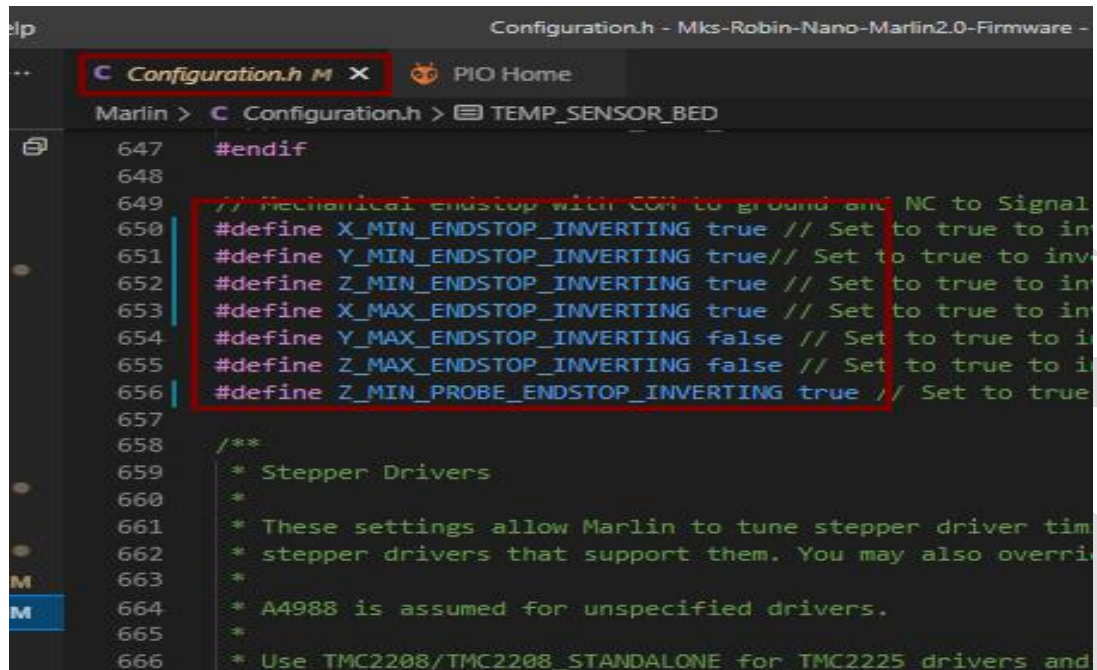
is hot bed



```
Configuration.h PIO Home
Marlin > C Configuration.h > TEMP_SENSOR_BED
459 * Use these for Testing or Development purposes
460 * 998 : Dummy Table that ALWAYS reads 25°C or the temperature
461 * 999 : Dummy Table that ALWAYS reads 100°C or the temperature
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
```

4.2.3 Enstop level type configuration

The configuration of the limit switch type (true/false), true is a normally open switch, and false is a normally closed switch.



```

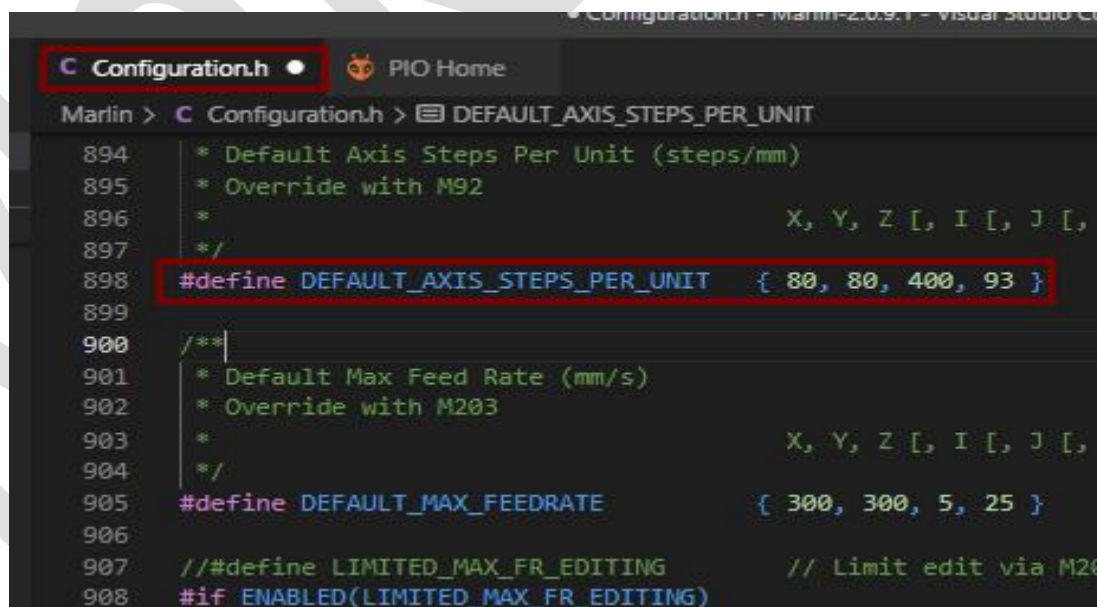
647 #endif
648
649 // Mechanical endstop with COM to ground and NC to Signal
650 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert
651 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert
652 #define Z_MIN_ENDSTOP_INVERTING true // Set to true to invert
653 #define X_MAX_ENDSTOP_INVERTING true // Set to true to invert
654 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert
655 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert
656 #define Z_MIN_PROBE_ENDSTOP_INVERTING true // Set to true to invert
657
658 /**
659  * Stepper Drivers
660  *
661  * These settings allow Marlin to tune stepper driver timing
662  * for stepper drivers that support them. You may also override
663  * these settings for drivers that do not support them.
664  * A4988 is assumed for unspecified drivers.
665  *
666  * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and

```

4.2.4 Pulse setting

#Define DEFAULT_AXIS_STEPS_PER_UNIT {80, 80, 400, 93} in the configuration file to set the pulses of the X, Y, Z, and E axes respectively.

The pulse value needs to be calculated and set according to the actual situation of the machine.



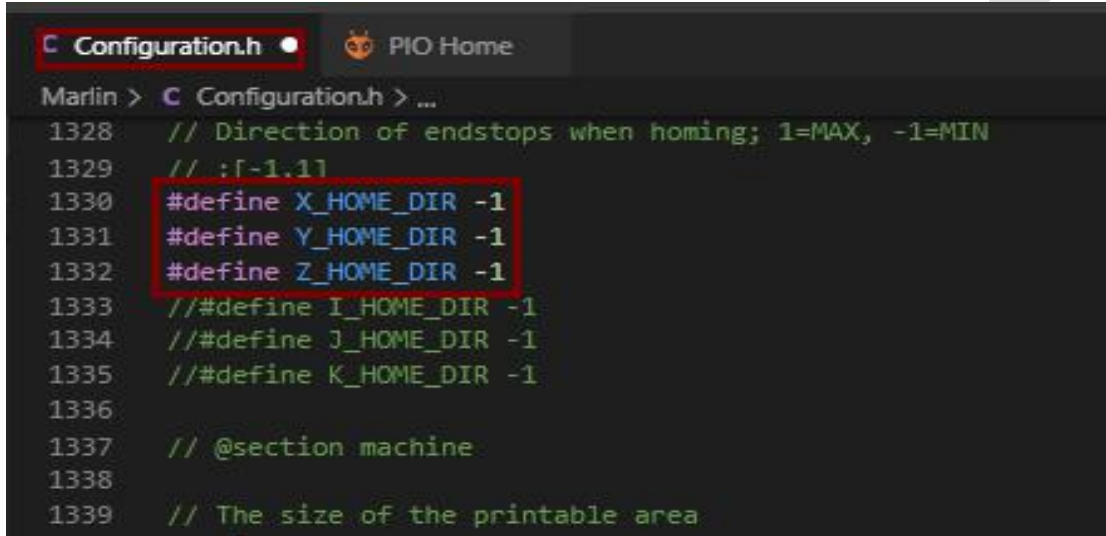
```

894 * Default Axis Steps Per Unit (steps/mm)
895 * Override with M92
896 *
897 * X, Y, Z [, I [, J [,
898 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 }
899
900 /**
901 * Default Max Feed Rate (mm/s)
902 * Override with M203
903 *
904 * X, Y, Z [, I [, J [,
905 #define DEFAULT_MAX_FEEDRATE { 300, 300, 5, 25 }
906
907 // #define LIMITED_MAX_FR_EDITING // Limit edit via M205
908 #if ENABLED(LIMITED_MAX_FR_EDITING)

```

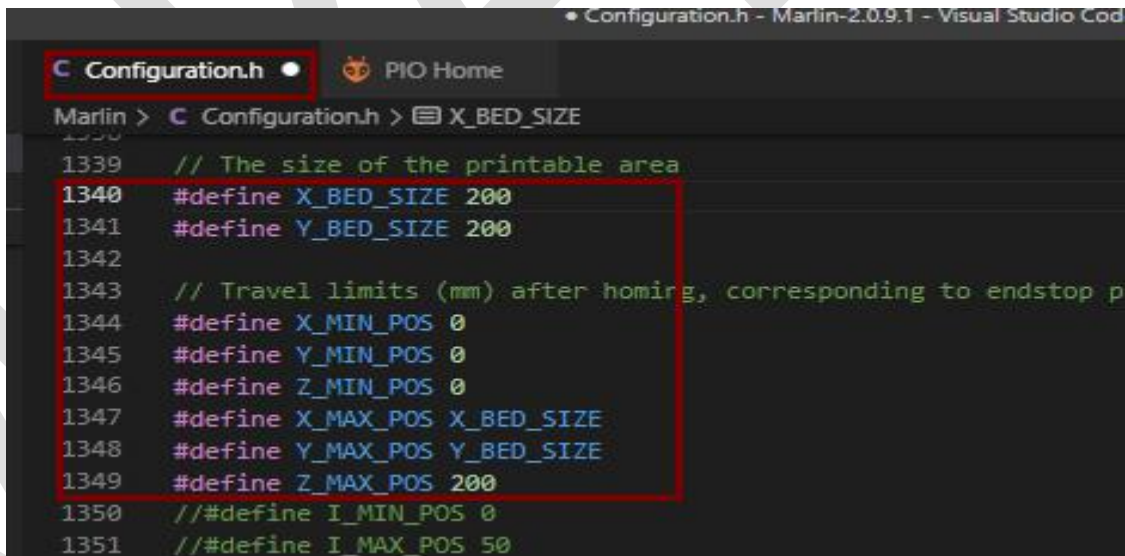
4.2.5 Zero return direction setting

Set the gohoming direction in the configuration file, -1 is the minimum direction, 1 is the maximum direction



```
Configuration.h PIO Home
Marlin > C Configuration.h > ...
1328 // Direction of endstops when homing; 1=MAX, -1=MIN
1329 // :[-1,1]
1330 #define X_HOME_DIR -1
1331 #define Y_HOME_DIR -1
1332 #define Z_HOME_DIR -1
1333 //#define I_HOME_DIR -1
1334 //#define J_HOME_DIR -1
1335 //#define K_HOME_DIR -1
1336
1337 // @section machine
1338
1339 // The size of the printable area
```

4.2.6 Print platform range setting

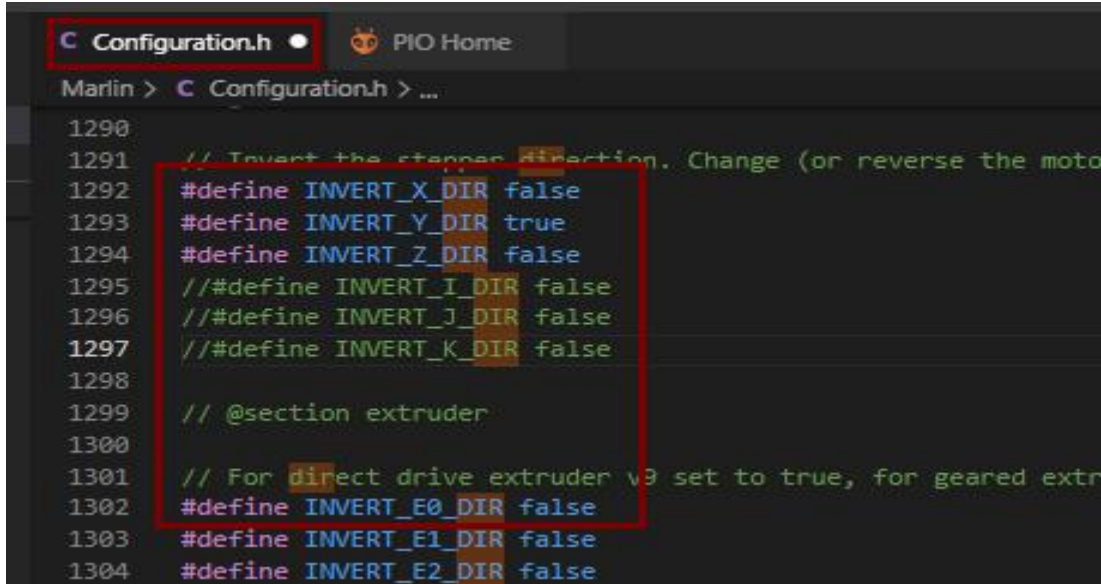


```
Configuration.h - Marlin-2.0.9.1 - Visual Studio Cod
PIO Home
Marlin > C Configuration.h > X_BED_SIZE
1339 // The size of the printable area
1340 #define X_BED_SIZE 200
1341 #define Y_BED_SIZE 200
1342
1343 // Travel limits (mm) after homing, corresponding to endstop p
1344 #define X_MIN_POS 0
1345 #define Y_MIN_POS 0
1346 #define Z_MIN_POS 0
1347 #define X_MAX_POS X_BED_SIZE
1348 #define Y_MAX_POS Y_BED_SIZE
1349 #define Z_MAX_POS 200
1350 //#define I_MIN_POS 0
1351 //#define I_MAX_POS 50
```

4.2.7 Motor direction setting

Motor direction setting, false and true represent two rotation directions,

when the moving direction is opposite, the opposite configuration is enough.



```
1290
1291 // Invert the stepper direction. Change (or reverse the motor direction)
1292 #define INVERT_X_DIR false
1293 #define INVERT_Y_DIR true
1294 #define INVERT_Z_DIR false
1295 // #define INVERT_I_DIR false
1296 // #define INVERT_J_DIR false
1297 // #define INVERT_K_DIR false
1298
1299 // @section extruder
1300
1301 // For direct drive extruder v9 set to true, for geared extruder set to false
1302 #define INVERT_E0_DIR false
1303 #define INVERT_E1_DIR false
1304 #define INVERT_E2_DIR false
```

4.3 Advanced configuration

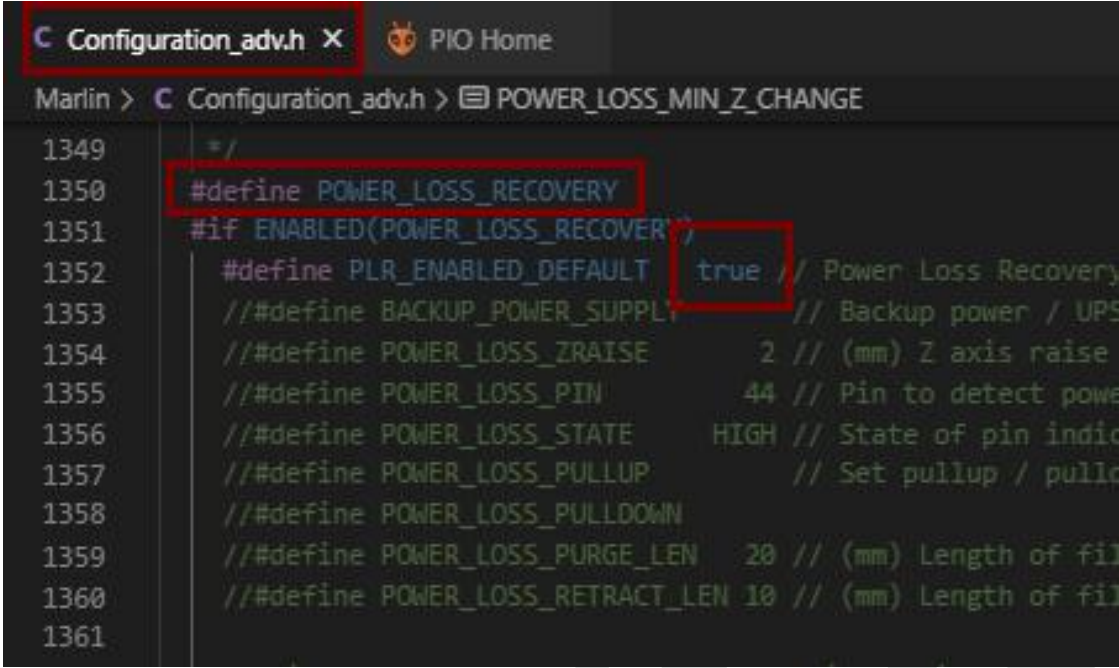
4.3.1 POWER_LOSS_RECOVERY

In the advanced configuration configuration_adv.h file, enable

#define POWER_LOSS_RECOVERY Change #define

PLR_ENABLED_DEFAULT to false

#Define PLR_ENABLED_DEFAULT true



```
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
1356   //#define POWER_LOSS_STATE HIGH // State of pin indicator
1357   //#define POWER_LOSS_PULLUP // Set pullup / pulldown
1358   //#define POWER_LOSS_PULLDOWN
1359   //#define POWER_LOSS_PURGE_LEN 20 // (mm) Length of filament
1360   //#define POWER_LOSS_RETRACT_LEN 10 // (mm) Length of filament
1361
```

4.3.2 FILAMENT_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 on start
1283 #define NUM_RUNOUT_SENSORS 1 // Number of sensors, up to 4
1284
1285 #define FIL_RUNOUT_STATE LOW // Pin state indicating that filament is out
1286 #define FIL_RUNOUT_PULLUP // Use internal pullup for filament out
1287 //#define FIL_RUNOUT_PULLDOWN // Use internal pulldown for filament out
1288 //#define WATCH_ALL_RUNOUT_SENSORS // Execute runout script on any sensor
1289 // | | | | | // This is automatically enabled when the sensor is
1290
1291 // Override individually if the runout sensors vary
1292 //#define FIL_RUNOUT1_STATE LOW
1293 //#define FIL_RUNOUT1_PULLUP
1294
1295

```

Set the level of the detection switch in the configuration file
(LOW/HIGH)

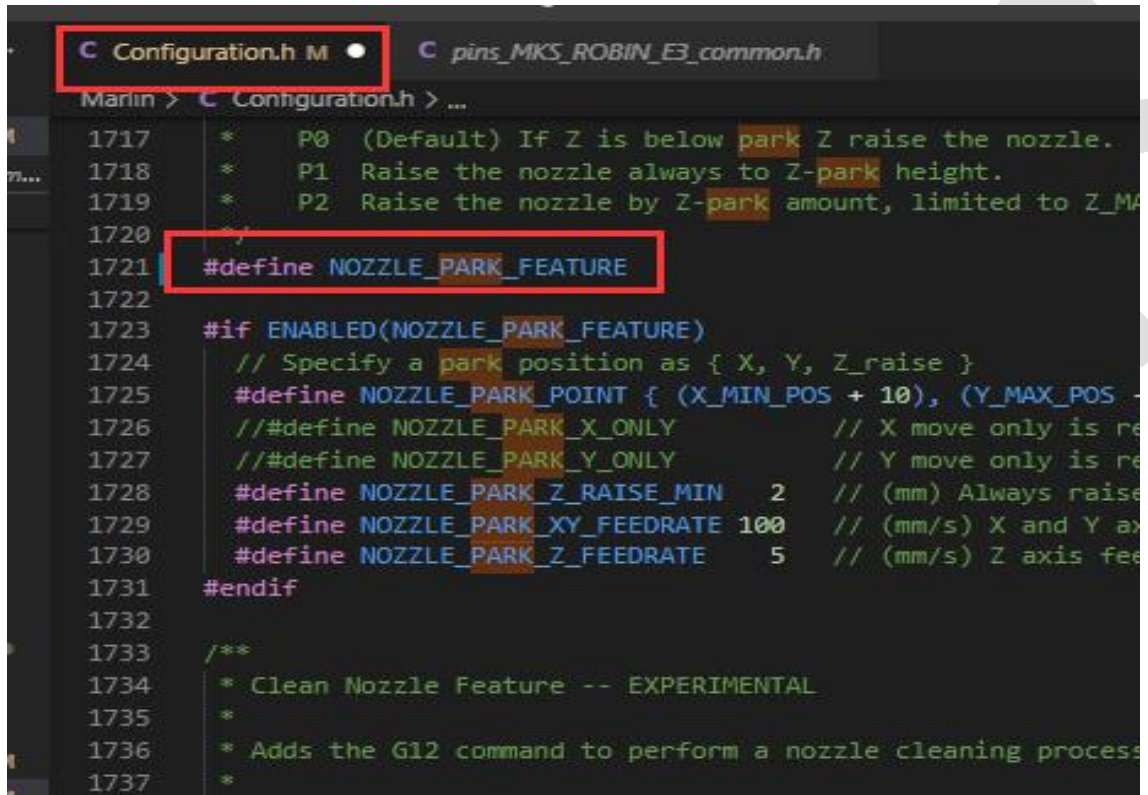
```

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 4
1284
1285 #define FIL_RUNOUT_STATE LOW // Pin state indicating that filament is out
1286 #define FIL_RUNOUT_PULLUP // Use internal pullup for filament out
1287 //#define FIL_RUNOUT_PULLDOWN // Use internal pulldown for filament out
1288 //#define WATCH_ALL_RUNOUT_SENSORS // Execute runout script on any sensor
1289 // | | | | | // This is automatically enabled when the sensor is
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
1295

```

Enable #define NOZZLE_PARK_FEATURE

in the configuration file



```
Configuration.h M pins_MKS_ROBIN_E3_common.h
Marlin > 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_M
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 -
1726 // #define NOZZLE_PARK_X_ONLY // X move only is re
1727 // #define NOZZLE_PARK_Y_ONLY // Y move only is re
1728 #define NOZZLE_PARK_Z_RAISE_MIN 2 // (mm) Always raise
1729 #define NOZZLE_PARK_XY_FEEDRATE 100 // (mm/s) X and Y ax
1730 #define NOZZLE_PARK_Z_FEEDRATE 5 // (mm/s) Z axis fee
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 advanced configuration file

```

Configuration_adv.h M
Marlin > C Configuration_adv.h > ADVANCED_PAUSE_FEATURE
2312 * Requires NOZZLE_PARK_FEATURE.
2313 * This feature is required for the default FILAMENT_RUNOUT_SC
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 shor
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 Bow
2324 // For dir
2325 // Set to
2326 #define FILAMENT_CHANGE_SLOW_LOAD_FEEDRATE 6 // (mm/s) S1
2327 #define FILAMENT_CHANGE_SLOW_LOAD_LENGTH 0 // (mm) Slow
    
```

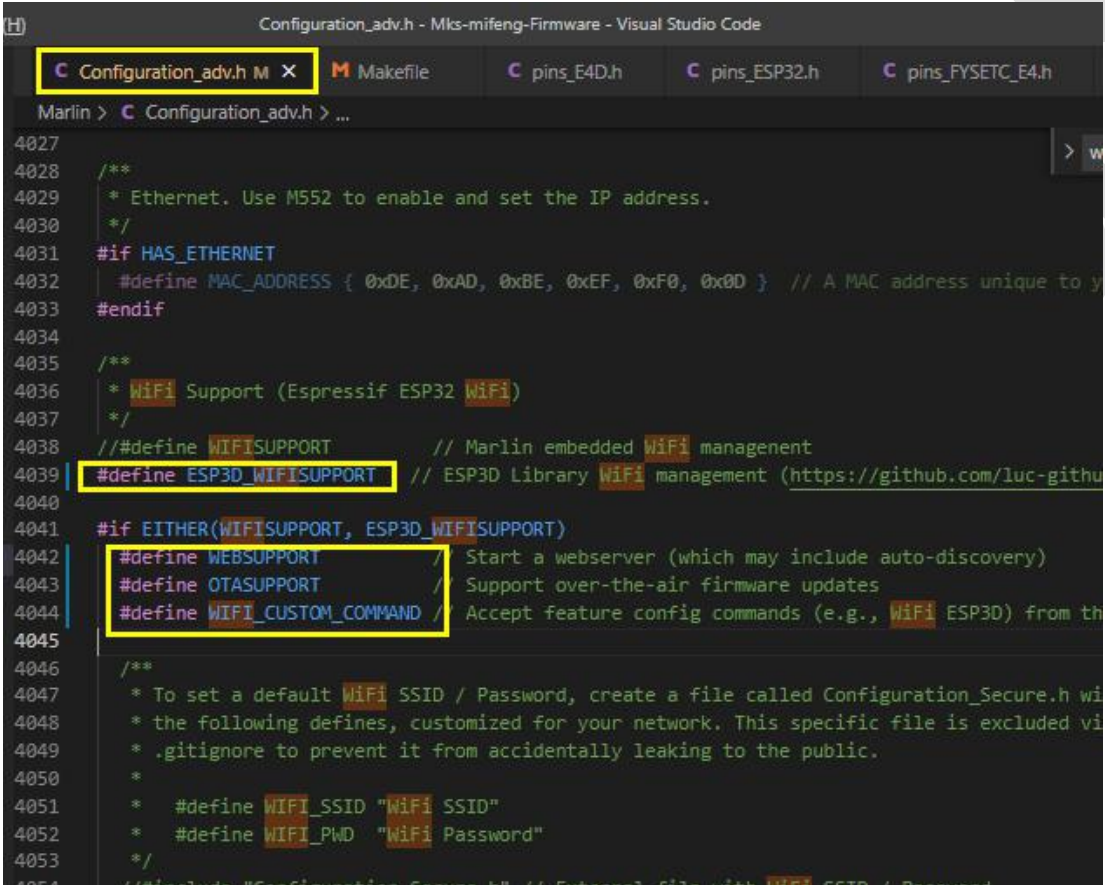
Enable #define PARK_HEAD_ON_PAUSE
in advanced configuration file

```

Configuration_adv.h M
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 // (secon
2350 #define FILAMENT_CHANGE_ALERT_BEEPS 10 // Number
2351 #define PAUSE_PARK_NO_STEPPER_TIMEOUT // Enable
2352
2353 #define PARK_HEAD_ON_PAUSE // Park the
2354 // #define HOME_BEFORE_FILAMENT_CHANGE // If nee
2355
2356 // #define FILAMENT_LOAD_UNLOAD_GCODES // Add M7
2357 // #define FILAMENT_UNLOAD_ALL_EXTRUDERS // Allow
2358 #endif
2359
2360 // Configuration
    
```


4.3.3 WIFI configuration

Enable `#define ESP3D_WIFISUPPORT` and `#define WEBSUPPORT`,
`#define OTASUPPORT`, `#define WIFI_CUSTOM_COMMAND` in
advanced configuration file

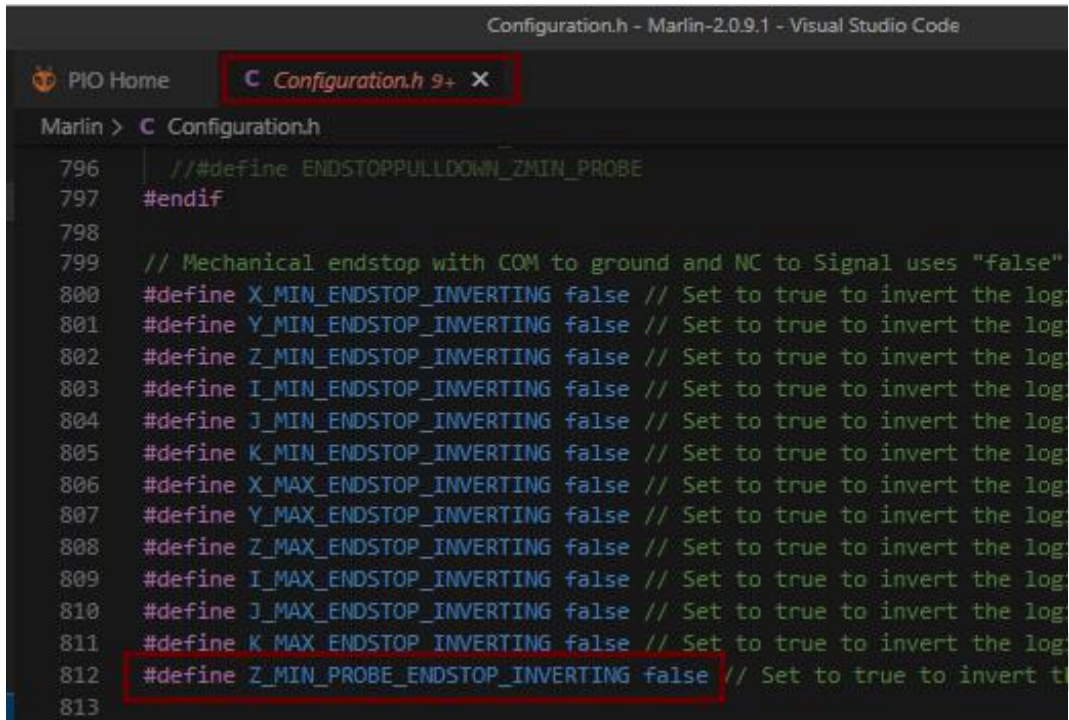


```
4027
4028 /**
4029  * Ethernet. Use M552 to enable and set the IP address.
4030  */
4031 #if HAS_ETHERNET
4032 #define MAC_ADDRESS { 0xDE, 0xAD, 0xBE, 0xEF, 0xF0, 0x0D } // A MAC address unique to you
4033 #endif
4034
4035 /**
4036  * WiFi Support (Espressif ESP32 WiFi)
4037  */
4038 // #define WIFISUPPORT // Marlin embedded WiFi management
4039 #define ESP3D_WIFISUPPORT // ESP3D Library WiFi management (https://github.com/luc-github/ESP3D)
4040
4041 #if EITHER(WIFISUPPORT, ESP3D_WIFISUPPORT)
4042 #define WEBSUPPORT // Start a webserver (which may include auto-discovery)
4043 #define OTASUPPORT // Support over-the-air firmware updates
4044 #define WIFI_CUSTOM_COMMAND // Accept feature config commands (e.g., WiFi ESP3D) from the
4045
4046 /**
4047  * To set a default WiFi SSID / Password, create a file called Configuration_Secure.h with
4048  * the following defines, customized for your network. This specific file is excluded via
4049  * .gitignore to prevent it from accidentally leaking to the public.
4050  *
4051  * #define WIFI_SSID "WiFi SSID"
4052  * #define WIFI_PWD "WiFi Password"
4053  */
4054 // #include "Configuration_Secure.h" // External file with WiFi SSID / Password
```

5. 3dtouch automatic leveling function

5.1 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
```

5.2 Set sensor signal pin

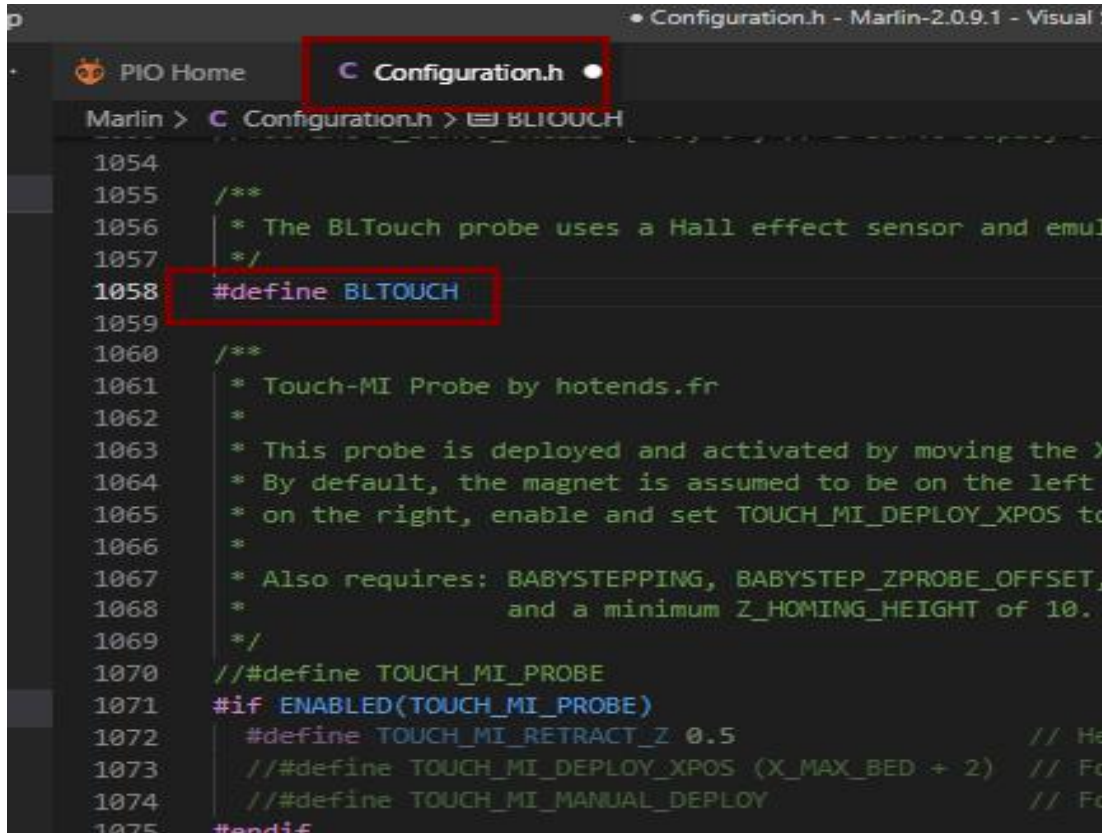
There is only z_min enstop interface on the motherboard, so z_safe_homing needs to be enabled when 3Dtouch is used for automatic leveling Port.

```

Configuration.h PIO Home pins.h pins_MKS_ESP_NANO.h
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
911 // Force the use of the probe for Z-axis homing
912 // #define USE_PROBE_FOR_Z_HOMING enable this item
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 pin:
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

```

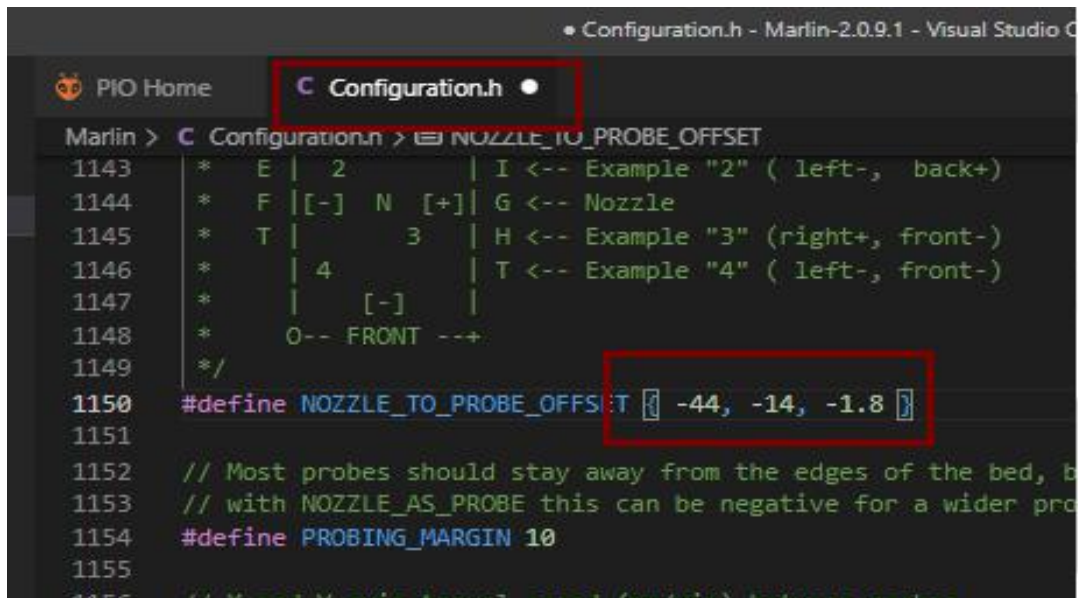
5.3 Enable BLTOUCH



```
1054
1055  /**
1056   * The BLTouch probe uses a Hall effect sensor and emul
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 // He
1073    //#define TOUCH_MI_DEPLOY_XPOS (X_MAX_BED + 2) // Fo
1074    //#define TOUCH_MI_MANUAL_DEPLOY // Fo
1075  #endif
```

5.4 Set the offset between the probe and the extrusion

They are the offsets of the X, Y, and Z axes. The offsets of X and Y need to be filled in according to the actual measurement. Z_offset can be tested and adjusted after leveling.



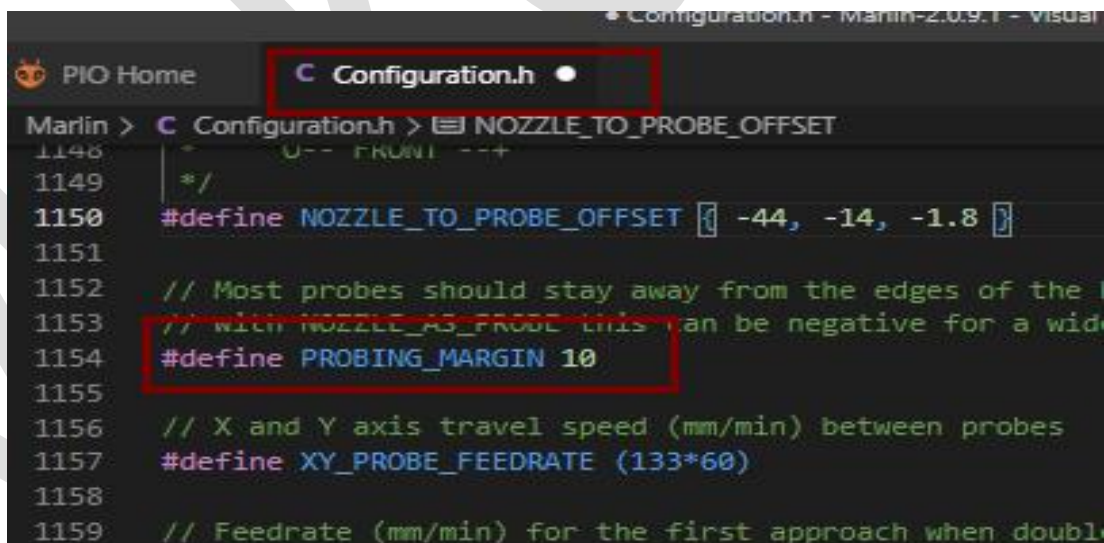
```

• Configuration.h - Marlin-2.0.9.1 - Visual Studio Code
PIO Home Configuration.h
Marlin > Configuration.h > NOZZLE_TO_PROBE_OFFSET
1143 * E | 2 | I <-- Example "2" ( left-, back+)
1144 * F | [-] N [+] | G <-- Nozzle
1145 * T | 3 | H <-- Example "3" (right+, front-)
1146 * | 4 | T <-- Example "4" ( left-, front-)
1147 * | [-] |
1148 * 0-- FRONT --+
1149 */
1150 #define NOZZLE_TO_PROBE_OFFSET [-44, -14, -1.8]
1151
1152 // Most probes should stay away from the edges of the bed, but
1153 // with NOZZLE_AS_PROBE this can be negative for a wider probe
1154 #define PROBING_MARGIN 10
1155
1156 // X and Y axis travel speed (mm/min) between probes
1157 #define XY_PROBE_FEEDRATE (133*60)
1158
1159 // Feedrate (mm/min) for the first approach when double

```

5.5 Set the distance between the sensor and the edge of the printing platform during leveling

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)



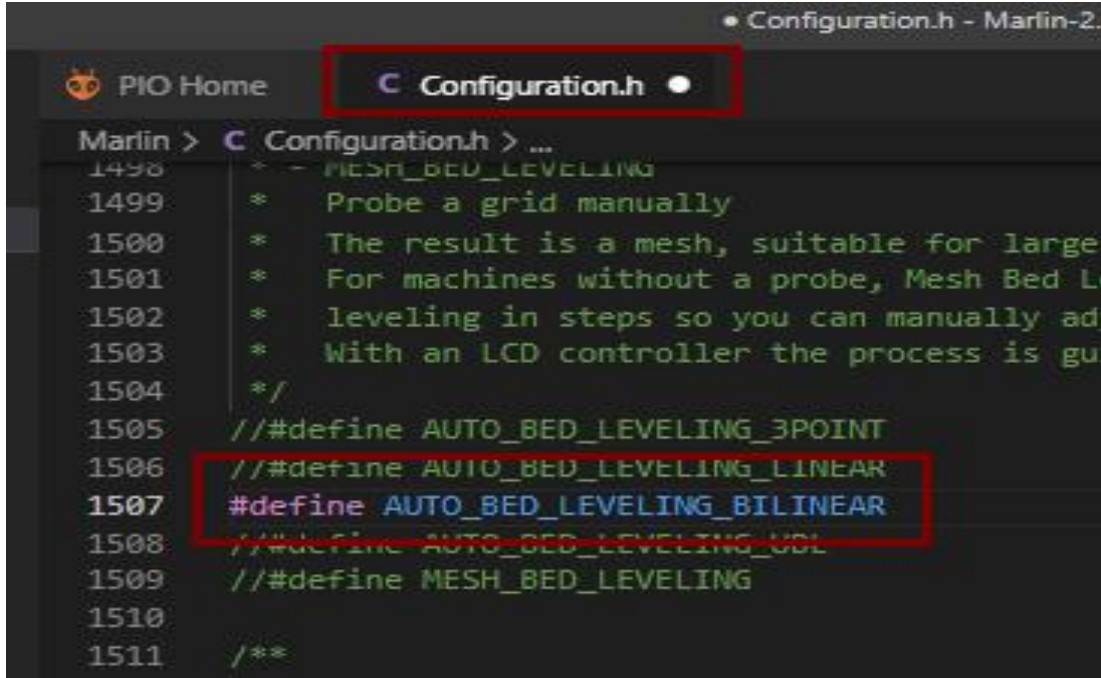
```

• Configuration.h - Marlin-2.0.9.1 - Visual Studio Code
PIO Home Configuration.h
Marlin > Configuration.h > NOZZLE_TO_PROBE_OFFSET
1146 * 0-- FRONT --+
1149 */
1150 #define NOZZLE_TO_PROBE_OFFSET [-44, -14, -1.8]
1151
1152 // Most probes should stay away from the edges of the bed, but
1153 // with NOZZLE_AS_PROBE this can be negative for a wider probe
1154 #define PROBING_MARGIN 10
1155
1156 // X and Y axis travel speed (mm/min) between probes
1157 #define XY_PROBE_FEEDRATE (133*60)
1158
1159 // Feedrate (mm/min) for the first approach when double

```


5.6 Enable automatic leveling

Enable linear auto leveling in the configuration file

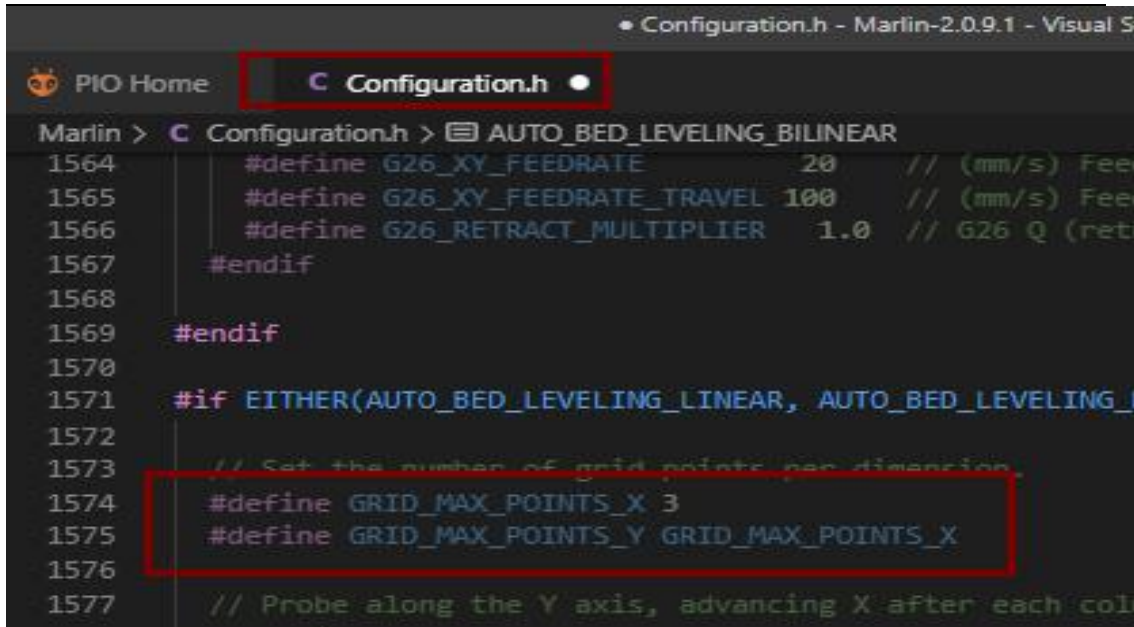


```
1498 * - MESH_BED_LEVELING
1499 *   Probe a grid manually
1500 *   The result is a mesh, suitable for large
1501 *   For machines without a probe, Mesh Bed L
1502 *   leveling in steps so you can manually ad
1503 *   With an LCD controller the process is gu
1504 */
1505 //#define AUTO_BED_LEVELING_3POINT
1506 //#define AUTO_BED_LEVELING_LINEAR
1507 #define AUTO_BED_LEVELING_BILINEAR
1508 //#define AUTO_BED_LEVELING_UDL
1509 //#define MESH_BED_LEVELING
1510
1511 /**
```

5.7 Set the number of grid points for leveling

Set the number of leveling points in the configuration file, the default is

3*3

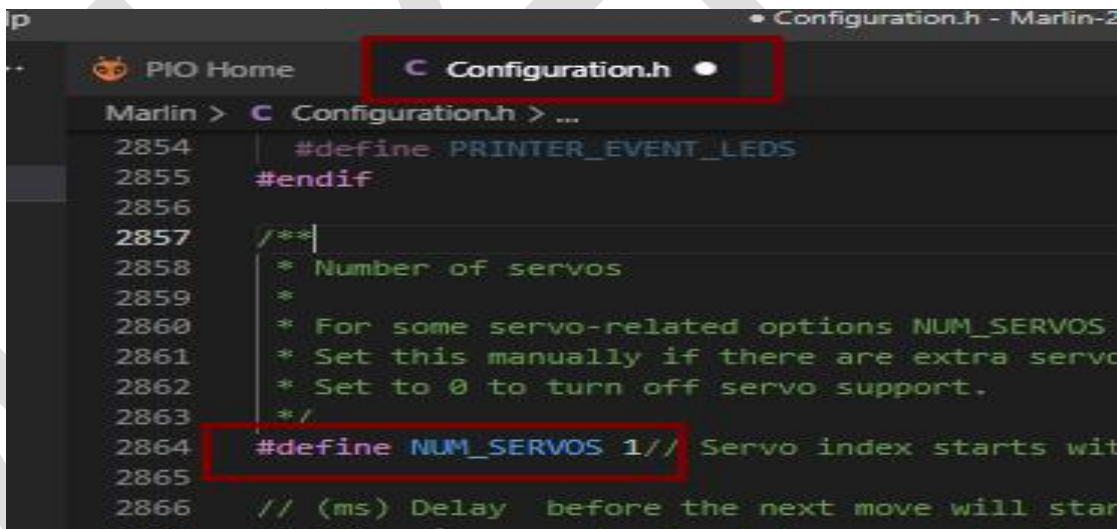


```

1564 #define G26_XY_FEEDRATE 20 // (mm/s) Fee
1565 #define G26_XY_FEEDRATE_TRAVEL 100 // (mm/s) Fee
1566 #define G26_RETRACT_MULTIPLIER 1.0 // G26 Q (ret
1567 #endif
1568
1569 #endif
1570
1571 #if EITHER(AUTO_BED_LEVELING_LINEAR, AUTO_BED_LEVELING_
1572
1573 // Set the number of grid points per dimension
1574 #define GRID_MAX_POINTS_X 3
1575 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
1576
1577 // Probe along the Y axis, advancing X after each col
    
```

5.8 Enable the servo

Enable the servo in the configuration file



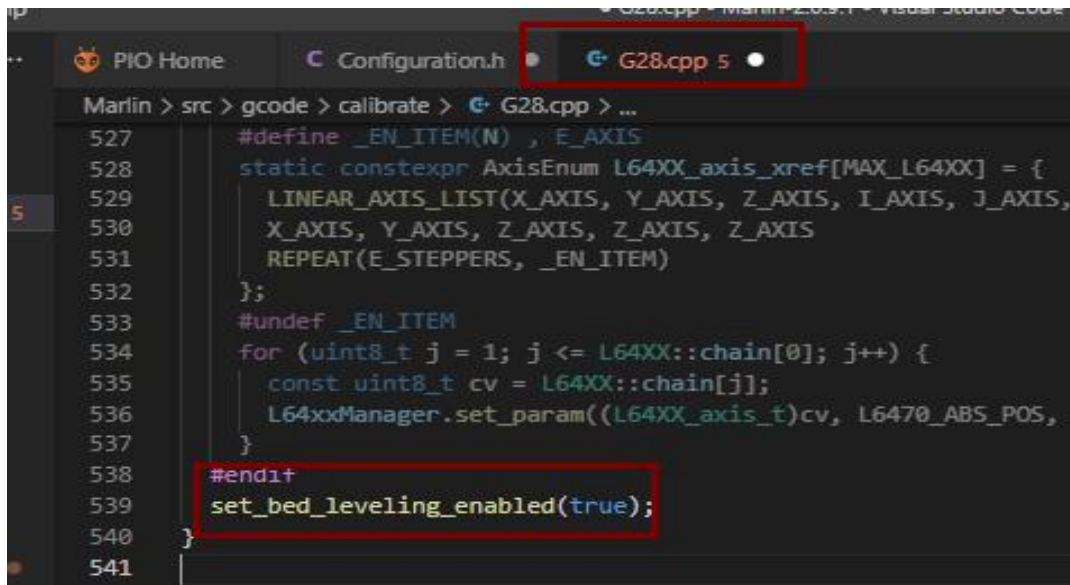
```

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
2865
2866 // (ms) Delay before the next move will start
    
```

5.9 Add auto-leveling data enable code

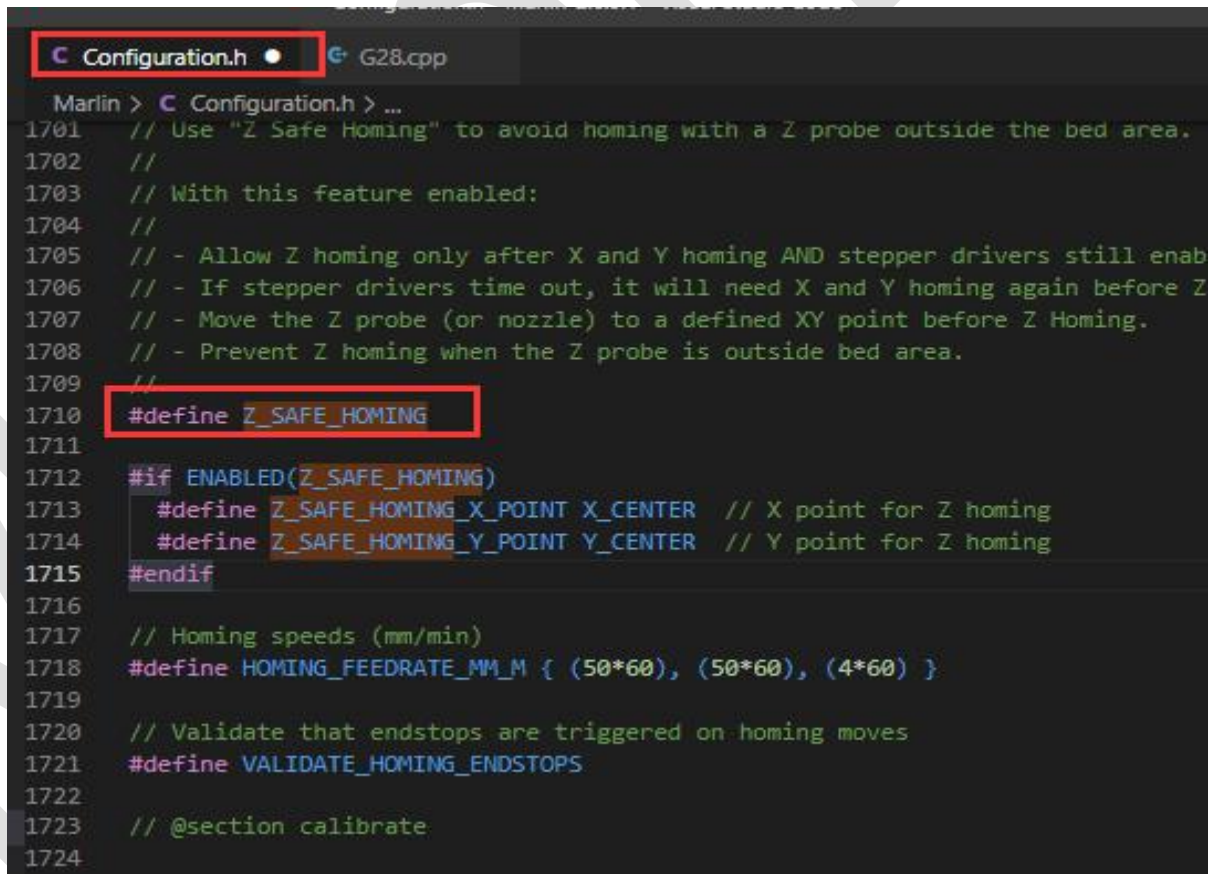
Added `set_bed_leveling_enabled(true);`

in G28.cpp file



```
Marlin > src > gcode > calibrate > G28.cpp > ...
527 #define _EN_ITEM(N) , E_AXIS
528 static constexpr AxisEnum L64XX_axis_xref[MAX_L64XX] = {
529     LINEAR_AXIS_LIST(X_AXIS, Y_AXIS, Z_AXIS, I_AXIS, J_AXIS,
530     X_AXIS, Y_AXIS, Z_AXIS, Z_AXIS, Z_AXIS)
531     REPEAT(E_STEPPERS, _EN_ITEM)
532 };
533 #undef _EN_ITEM
534 for (uint8_t j = 1; j <= L64XX::chain[0]; j++) {
535     const uint8_t cv = L64XX::chain[j];
536     L64xxManager.set_param((L64XX_axis_t)cv, L6470_ABS_POS,
537 }
538 #endif
539 set_bed_leveling_enabled(true);
540 }
541
```

5.10 enable z_safe_homing



```
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 enab
1706 // - If stepper drivers time out, it will need X and Y homing again before Z
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
```

6.WEB connection and settings

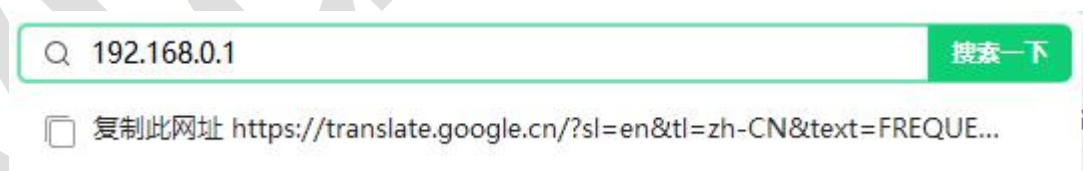
The firmware has enabled wifi. After the motherboard updates the firmware, the default wifi is AP mode (local area network). At this time, use a computer or mobile phone to view the wifi list, and you can see that the name of the wifi is MARLIN_ESP.



1. Enter the password to connect to wif (the default wifi password is 12345678);

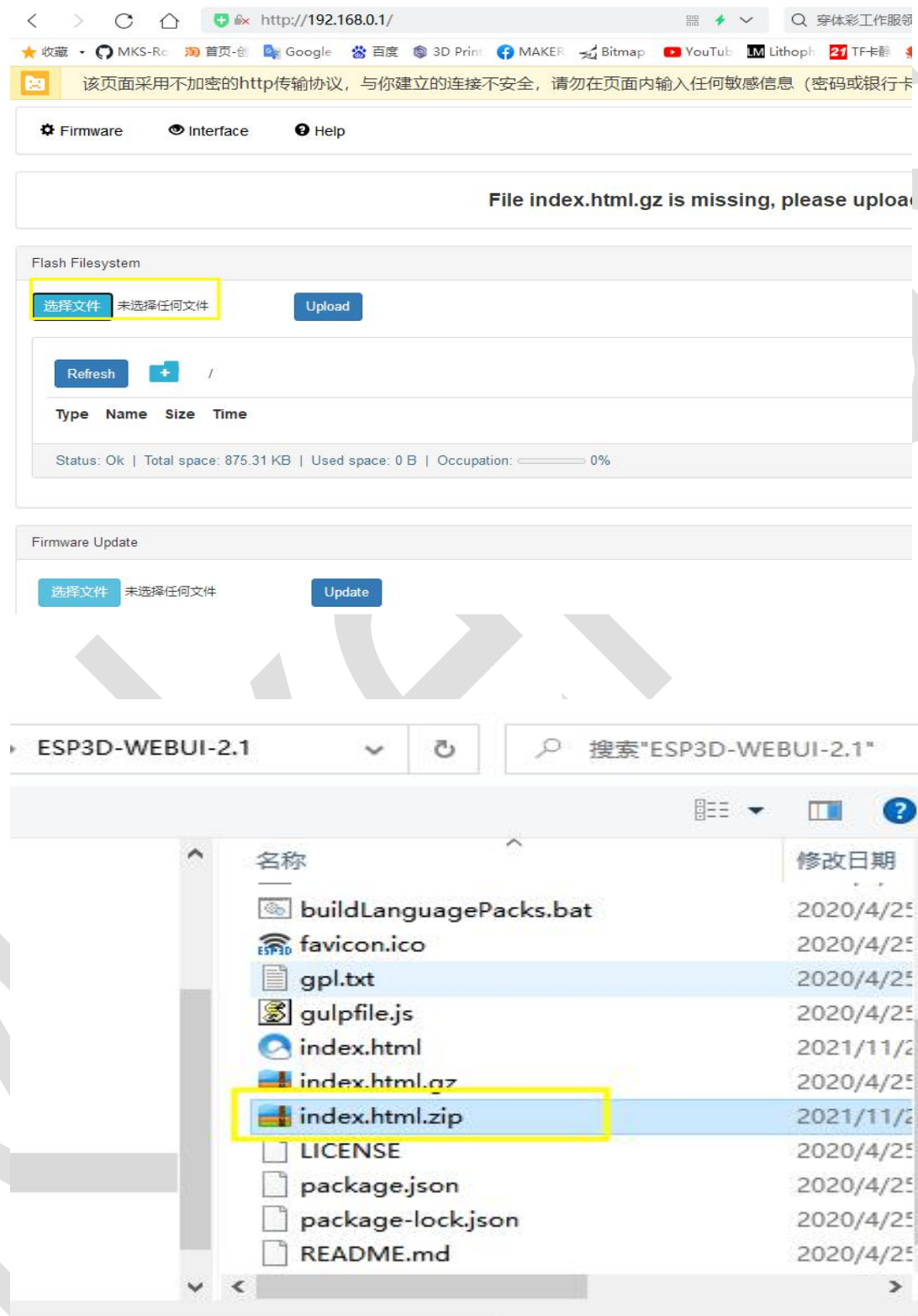


2. in the browser, enter the ip to log in to the web interface (the default ip address is 192.168.0.1);



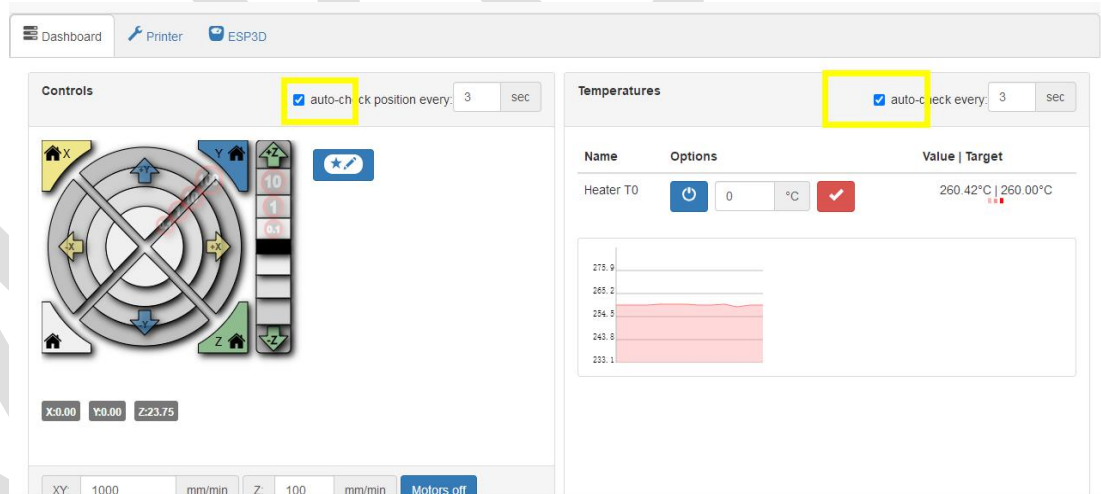
3. Then click to upload the file on the web control interface, the file

download link:<https://github.com/luc-github/ESP3DLib>



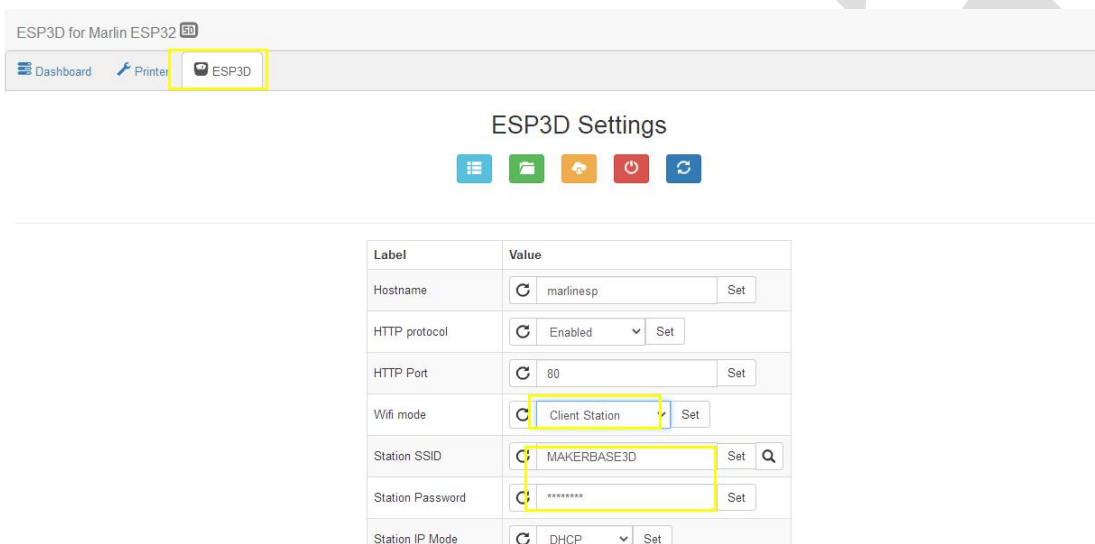


4. After uploading, check the refresh time on the web page, and you can start operation control after seeing the temperature



5. STA mode setting, enter the ESP3D interface, then select Client

Station, enter the name and password of the wifi you want to connect, check set, and then need to restart the motherboard. If you are using LCD, you can view it on the LCD interface To the assigned ip, if you use a serial port screen or a useless screen, you need to log in to the router's management page to view the ip; then re-enter the ip on the browser to enter the web.



7、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.

Makerbase