



Makerbase

Guangzhou Qianhui Information technology Co., Ltd.

SGEN_L Motherboard Datasheet

Makerbase

QQ Group : 489095605 232237692

Email : Huangkaida@makerbase.com.cn


Document Version 1.0

Release Date: June 11th, 2019

The copyright belongs to Guangzhou Qianhui Information Technology., Ltd. All Rights Reserved

No unit or individual may extract or reproduce part or all of the contents of this document and shall disseminate it in any forms without the written permission of the company.

Trademark Statement

 Makerbase or “Makerbase” are trademarks of Guangzhou Qianhui Information Technology Co., Ltd. All other trademarks or registered trademarks mentioned in this document are the property of their respective owners.

Attention

The products, services or characteristics you purchase should be restricted to the Commercial Contracts and Terms of Guangzhou Qianhui Information Technology Co., Ltd. and all or part of the products, services or characteristics described in this document may not be included in the scope of your purchase or use. Unless otherwise agreed the contract, Guangzhou Qianhui Information Technology Co., Ltd. makes no representations or warranties, express or implied, regarding the contents of this document. The contents of this document may be updated irregularly due to product version upgrades or other reasons. Unless otherwise agreed, this document is only provided as a guide, and all statements, information, and recommendations in this document are not warranties of any kind, express or implied.

Version Update

Version	Modification time	Modification	Remark
V1.0	11 th June, 2019	Original version	

Makerbase

Catalogue

I Overview.....	5
II Features.....	6
III Wiring and size diagram.....	7
1, MKS SGen-L.....	7
2、 MKS SGen_L wiring.....	7
3、 MKS SGen_L installation diagram.....	8
4、 MKS SGen_L pin interface diagram.....	8
IV Instruction.....	9
1,The way to get firmware.....	9
2. The way to update firmware.....	9
V Parameter configuration instruction.....	16
5.1 Smoothieware parameter configuration.....	16
5.2 Marlin2.0 parameter setting.....	22
5.2.1 Basic parameter setting.....	22
5.2.2 Automatic Leveling Setting.....	32
5.2.3 Driver mode setting.....	37
Normal mode (step/dir mode).....	37
TMC2130 SPI mode setting.....	38
UART mode setting (MKS TMC2209, MKS TMC2208 V2.0).....	43
VI Printing connection and driver installation.....	47
VII Technical support and guarantee.....	49

I Overview

MKS-SGen_L is a 3D printer ARM motherboard developed by Makerbase. It uses a 32-bit Cortex-M3 LPC 1768 processor, whose running frequency is up to 100 MHz. It is compatible with Smoothieware and Marlin 2.0 firmware. It can be used with external and direct-plug drivers, in which, direct-plug driver is compatible with various modes (uart, spi, etc.). It can be connected to multiple LCD screens and MKS series of touch screens. It is a type of motherboard with good performance, high quality and fair price.

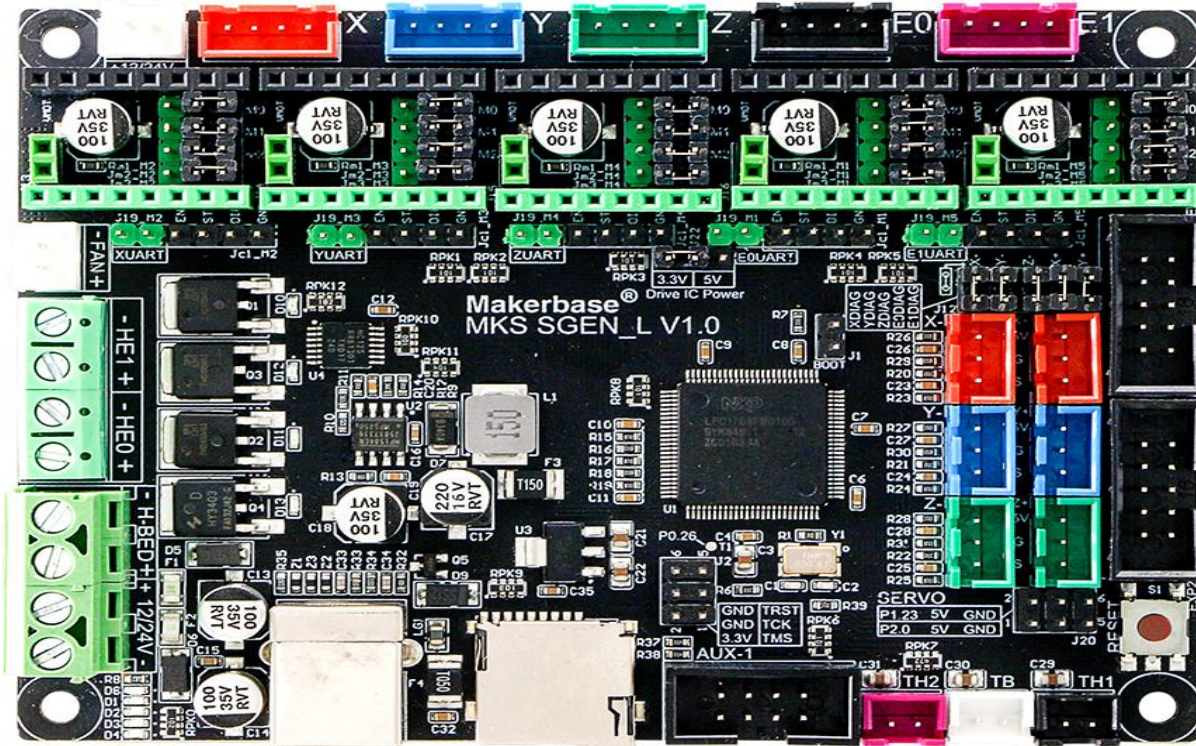


II Features

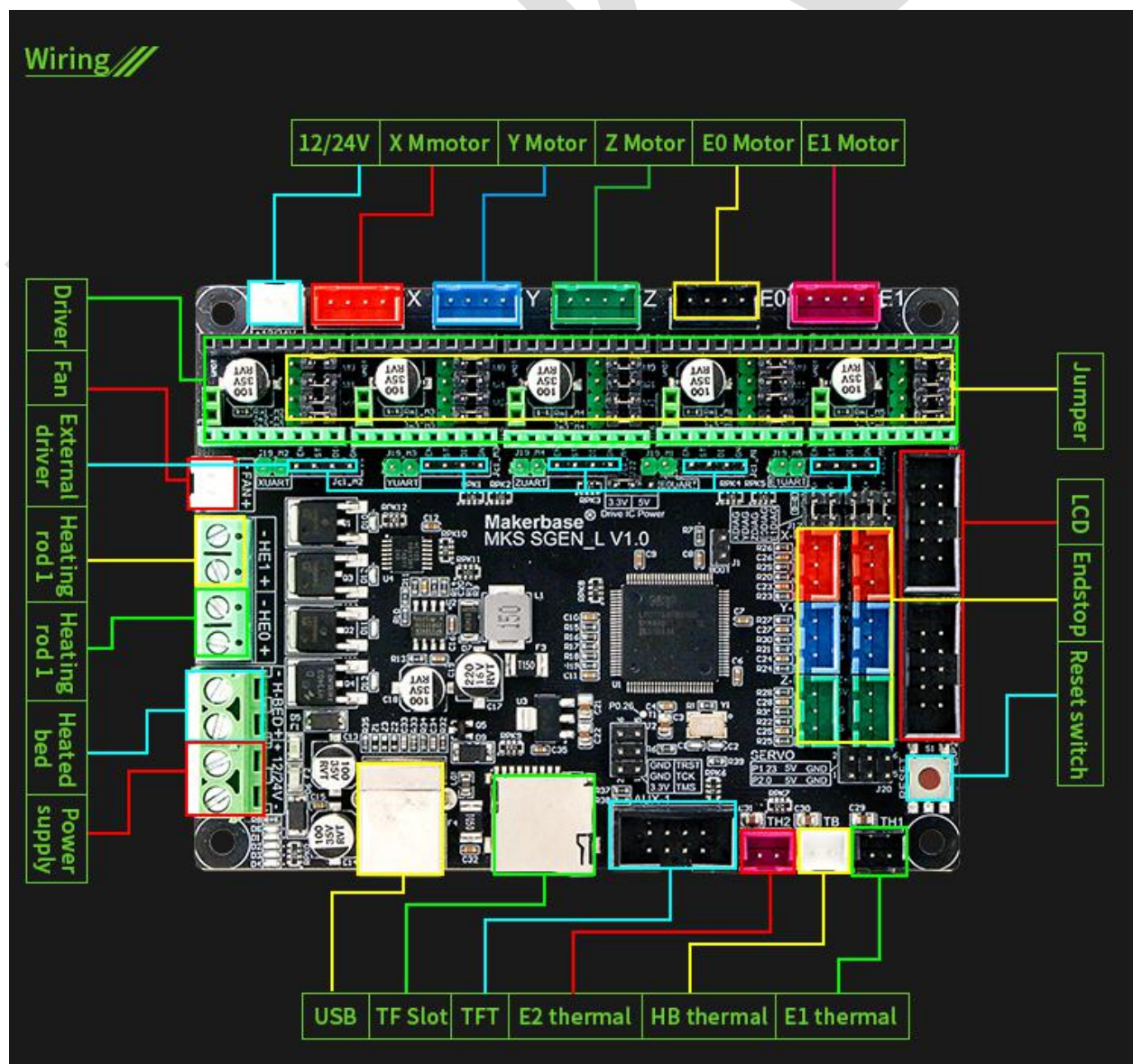
1. Adopt 32-bit Cortex-M3 LPC 1768 chip, 100MHZ frequency, strong computing power, performance greatly improved.
2. In-line and external driver port, more expandable and selective.
3. Compatible with multiple drivers and special modes (TMC2208 UART mode, TMC2130 SPI mode, etc.), dedicated port reservation, convenient for wiring settings .
4. Support lots of open-source firmwares, including smoothie-ware and Marlin V2.0.
5. Use high-quality PCB, and the corresponding pins of each interface are marked with silk screen for easier identification and modification.
6. Using a dedicated power chip, support 12V-24V current input.
7. Support the MKS LCD12864, MKS MINI12864 and other controller boards developed by Makerbase.
8. Compatible with all of MKS TFT touch screens.

III Wiring and size diagram

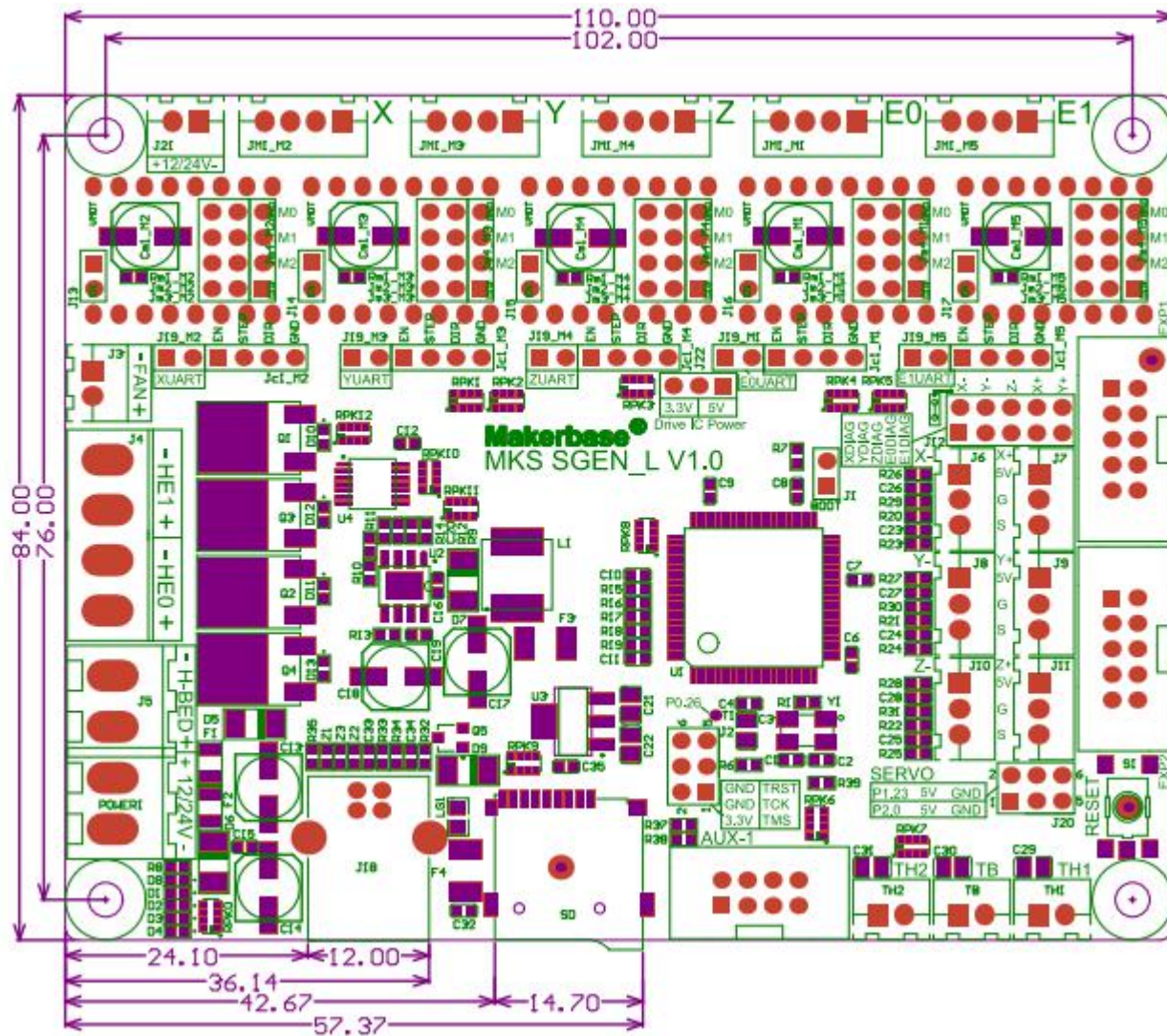
1, MKS SGen-L



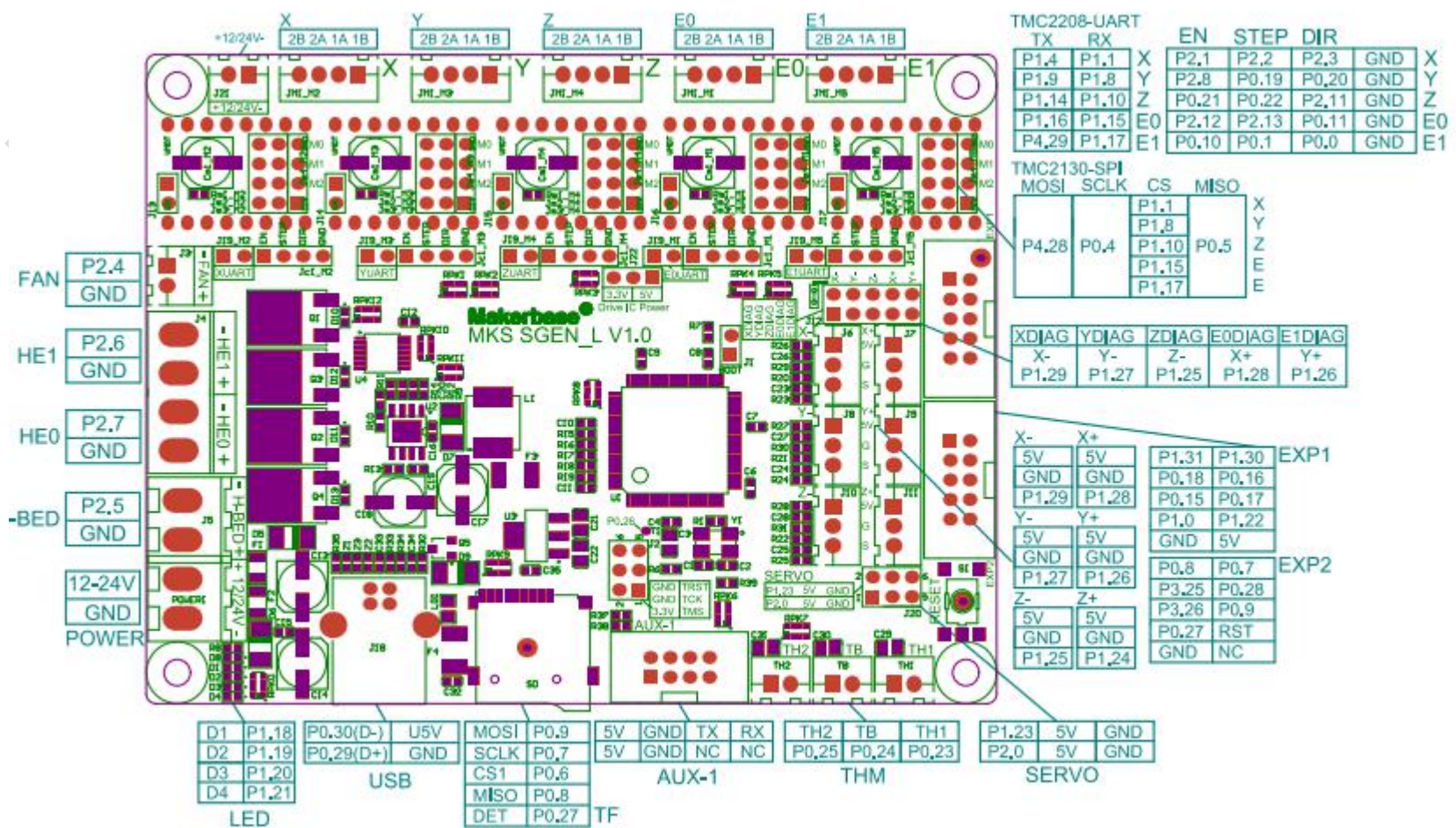
2, MKS SGen_L wiring



3、MKS SGen_L installation diagram



4、MKS SGen_L pin interface diagram



IV Instruction

1,The way to get firmware

- 1.1 Get firmware from customer service staffs or technicians
- 1.2 Download firmware on Makerbase QQ group
- 1.3 Website: <https://github.com/makerbase-mks?tab=repositories>

2. The way to update firmware

Including smoothieware and Malin

2.1 Smoothieware update

2.1.1 Copy the update program file to the TF card root directory, including:

- 1. firmware.bin
- 2. file config.txt

2.1.3 Insert the TF card into the card slot and power it on again. The new firmware will be upgraded automatically.

After the update is successful, the SD card file will be renamed to “firmware.cur”.

2.1.3 After updating the firmware, the indicator in the upper left corner of the motherboard is normally D1 D4 D7 is always on, and D2 D3 is blinking;

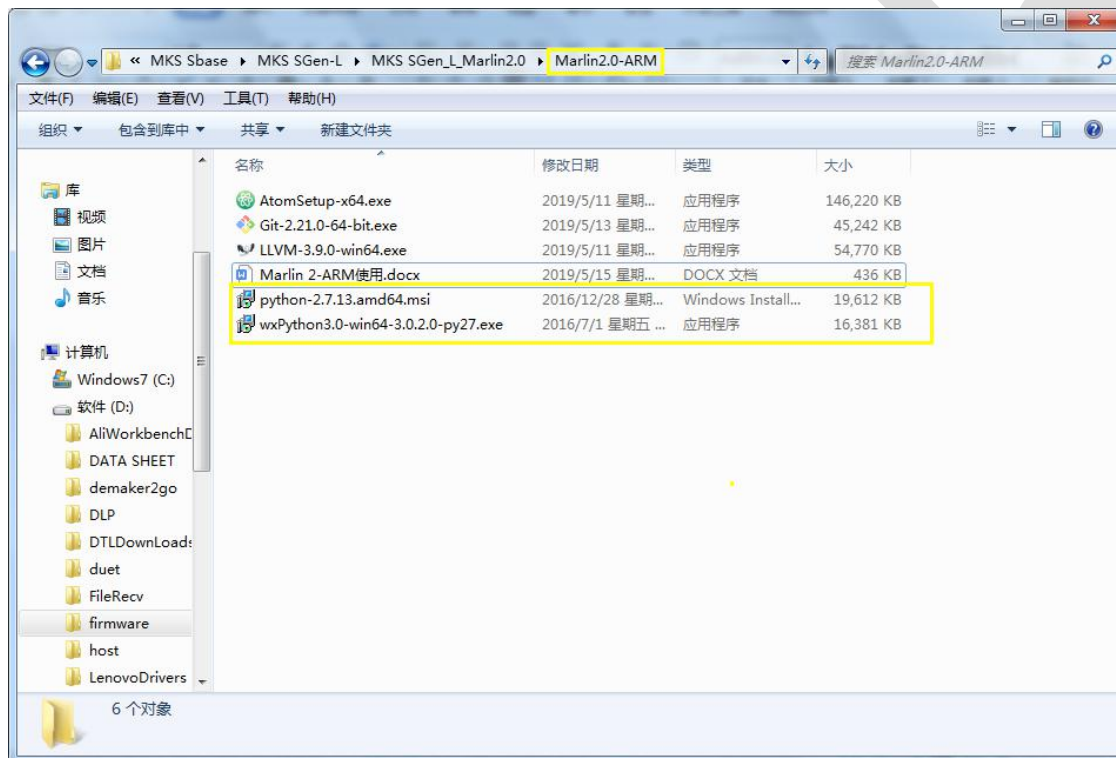
2.2 Marlin2.0 update

The way to update Marlin 2.0 firmware is similar to smoothieware. Copy the firmware.bin file to TF card and update the motherboard. After updating, The file will become the firmware.CUR file.

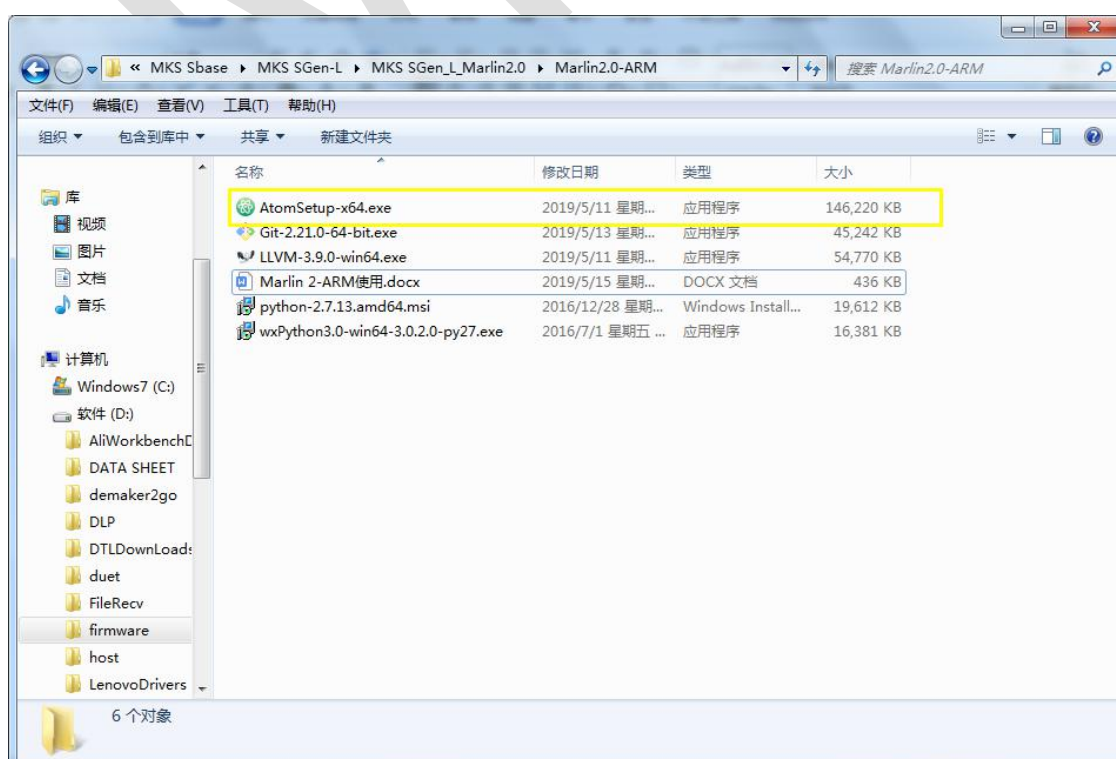
Not the same as smoothieware, marlin 2.0 firmware configuration cannot be modified via “config file”. updating parameter configuration of marlin firmware, must create a new “firmware.bin”file first.

2.2.1 Compilation environment built and “bin file” creation

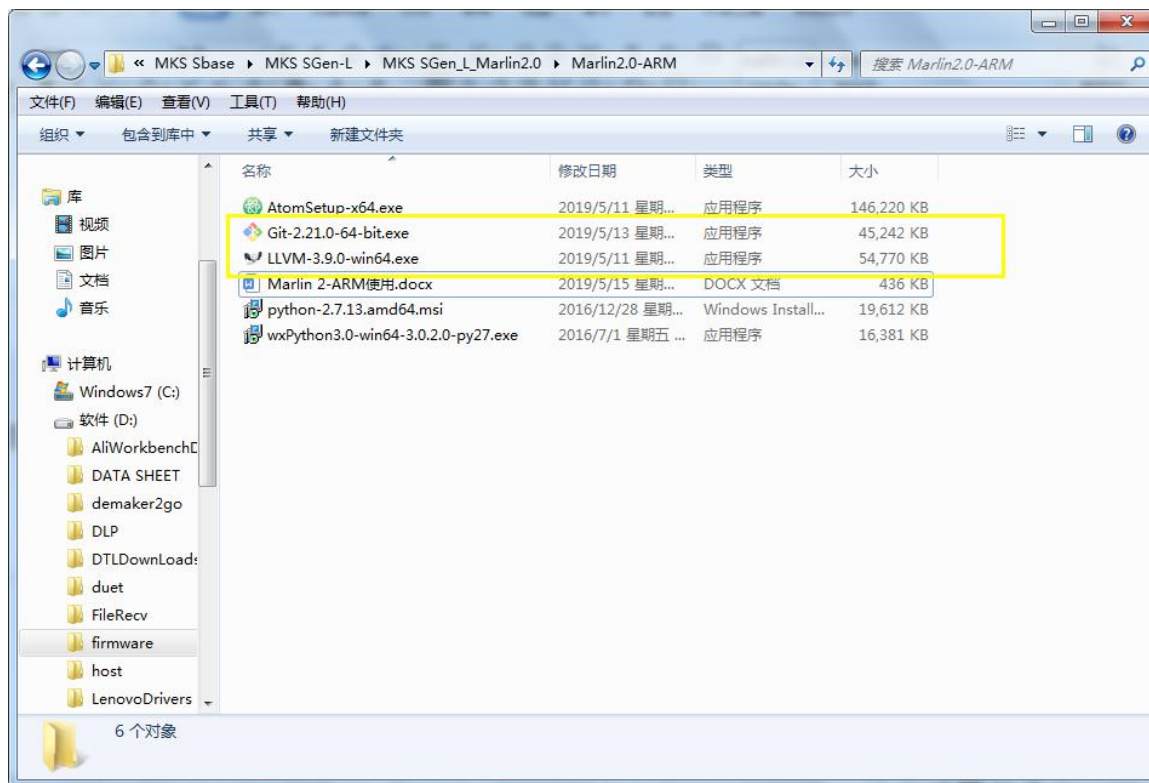
1. Download zipped file of firmware and software from website.
2. Open “marlin2.0-ARM” folder, install “python-2.7” and “wxPython3.0”. click on the zipped file to install, as the following figure shows



3. Atom installation



4. Install Git-2.21.0 and LLVM (only support 3.9.0)

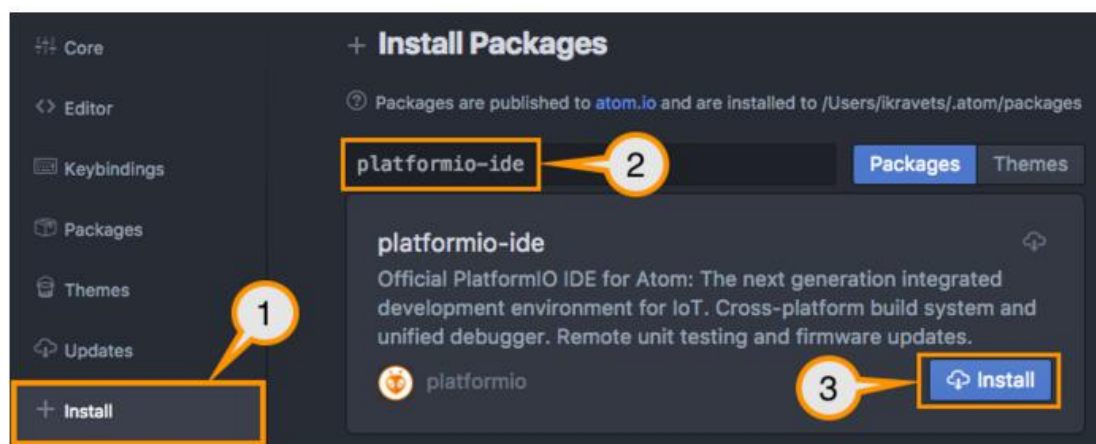
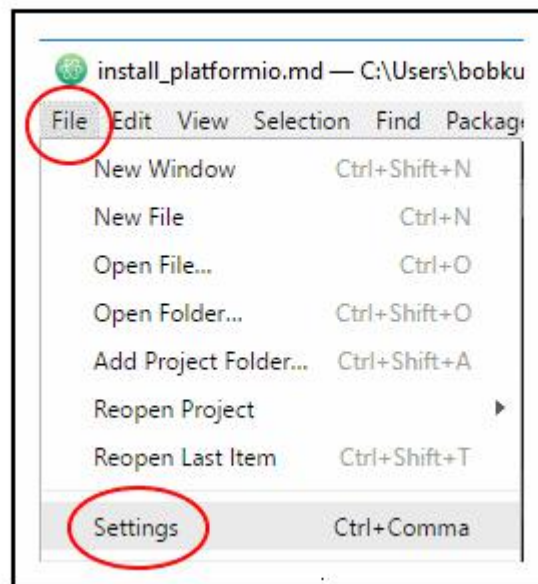


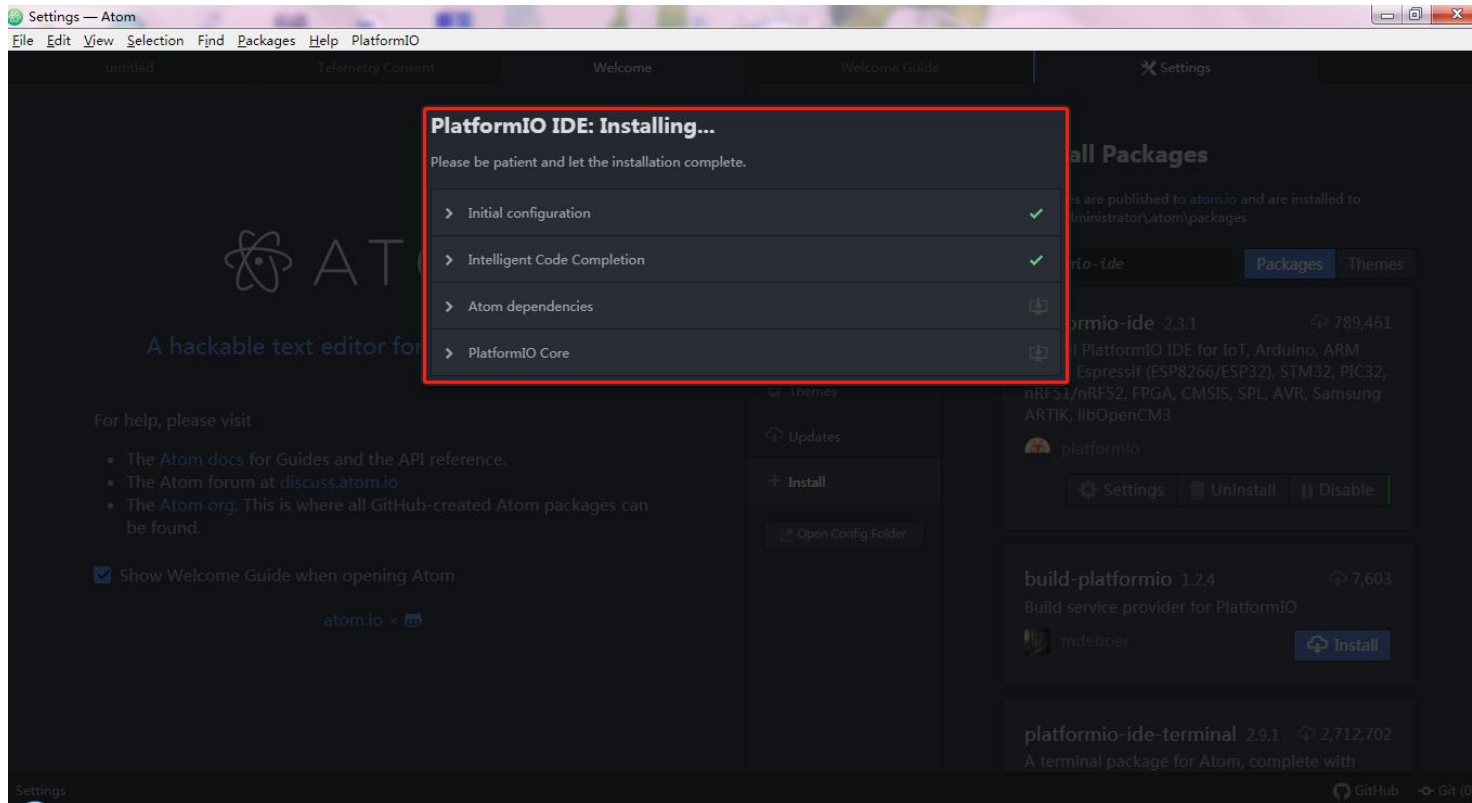
5. Atom software and plug-in installation

After installing Atom, open Atom, open File→Settings→install, it needs to install 2 plug-ins

Search “platformio_ide” to install.

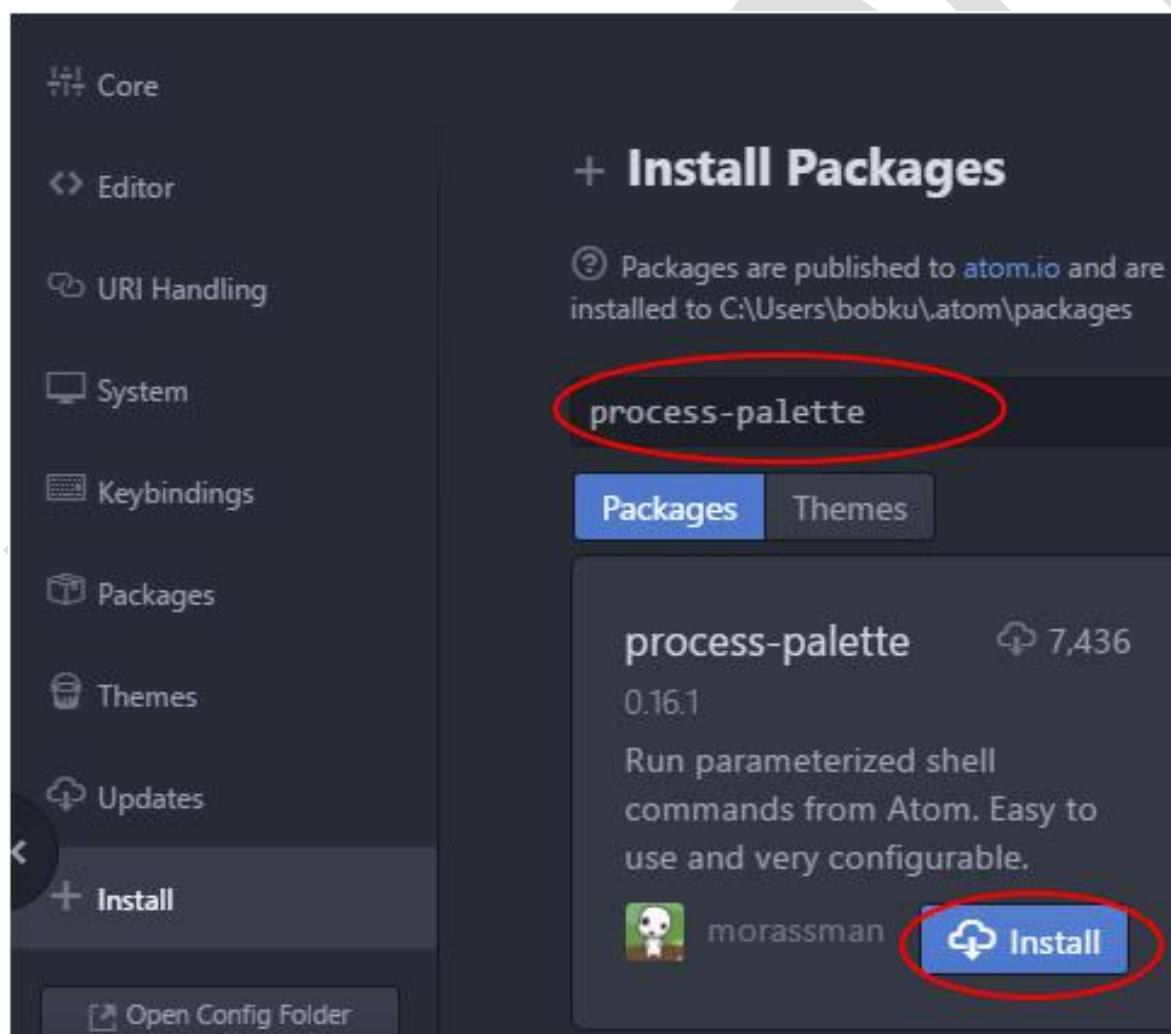
Windows: Click File then click Settings





Waiting for installation to complete

Search “process-palette” to install.



After the installation is completed, import the marlin 2.0 source code for modification and compilation.

After that, the compilation environment is built.

6. Compilation of “Bin” file

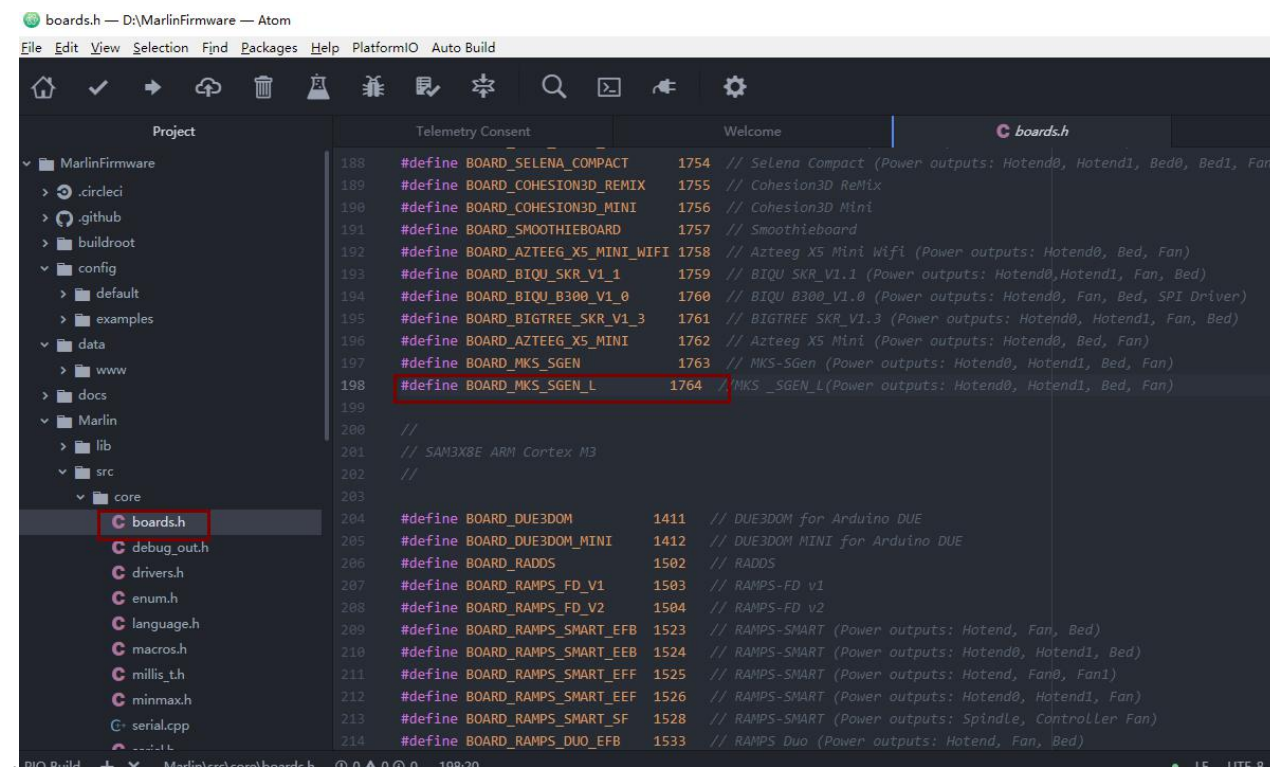
1, the input of firmware engineering

(The file path of the Marlin2.0 firmware cannot be Chinese, the recommended path is not too long, otherwise the compilation error) Click File→Open Folder to open the marlin2.0 firmware folder and import the marlin firmware.

2, motherboard model definition

Open "board.h", the motherboard type that can find MKS_SGEN_L is defined as "BOARD_MKS_SGEN_L".

Download the version of marlin2.0 and find the corresponding motherboard type in the board.h file



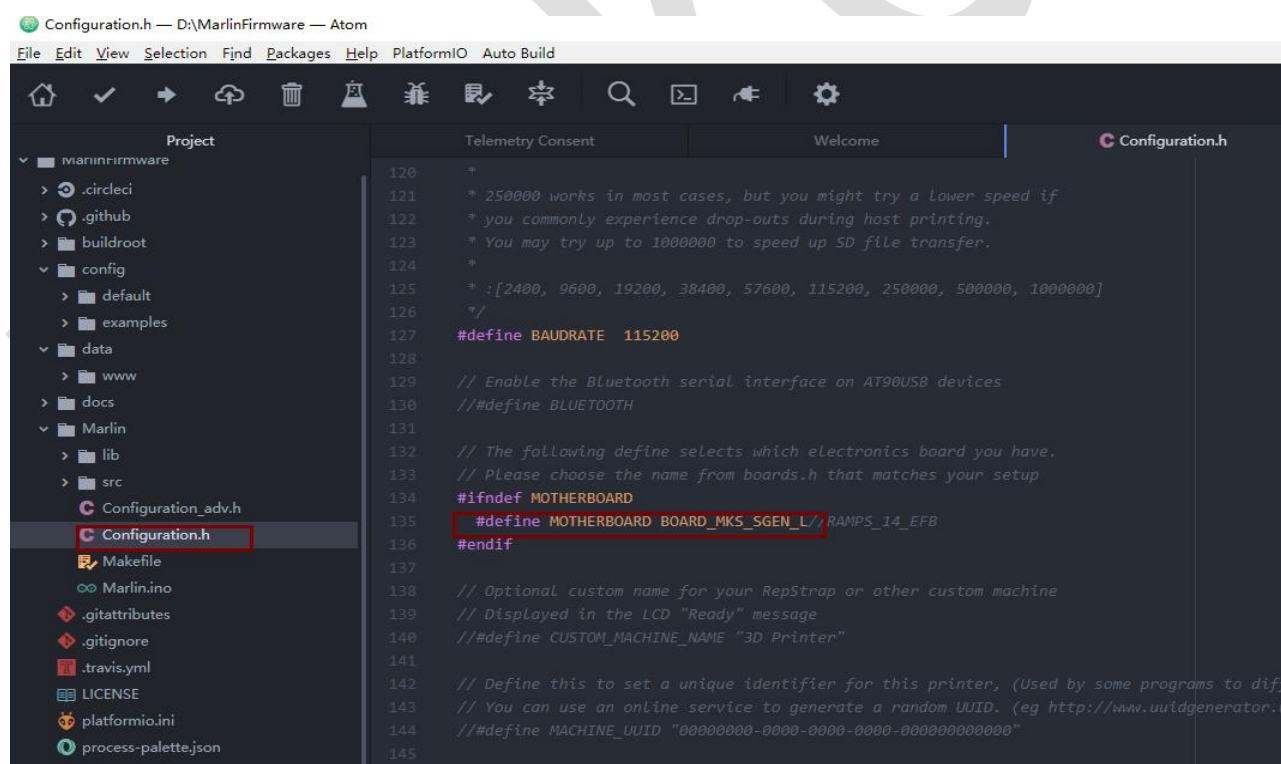
```

boards.h — D:\MarlinFirmware — Atom
File Edit View Selection Find Packages Help PlatformIO Auto Build

Project: MarlinFirmware
  .circleci
  .github
  buildroot
  config
  default
  examples
  data
  www
  docs
  Marlin
  lib
  src
  core
  boards.h
  debug_out.h
  drivers.h
  enum.h
  language.h
  macros.h
  millis_th
  minmax.h
  serial.cpp

Telemetry Consent Welcome boards.h
188 #define BOARD_SELENA_COMPACT 1754 // Selena Compact (Power outputs: Hotend0, Hotend1, Bed0, Bed1, Fan0, Fan1)
189 #define BOARD_COHESION3D_REMIX 1755 // Cohesion3D ReMix
190 #define BOARD_COHESION3D_MINI 1756 // Cohesion3D Mini
191 #define BOARD_SMOOTHIEBOARD 1757 // Smoothieboard
192 #define BOARD_AZTEEG_X5_MINI_WIFI 1758 // Azteeg X5 Mini Wifi (Power outputs: Hotend0, Bed, Fan)
193 #define BOARD_BIQU_SKR_V1_1 1759 // BIQU_SKR_V1.1 (Power outputs: Hotend0, Hotend1, Fan, Bed)
194 #define BOARD_BIQU_B300_V1_0 1760 // BIQU_B300_V1.0 (Power outputs: Hotend0, Fan, Bed, SPI Driver)
195 #define BOARD_BIGTREE_SKR_V1_3 1761 // BIGTREE_SKR_V1.3 (Power outputs: Hotend0, Hotend1, Fan, Bed)
196 #define BOARD_AZTEEG_X5_MINI 1762 // Azteeg X5 Mini (Power outputs: Hotend0, Bed, Fan)
197 #define BOARD_MKS_SGEN 1763 // MKS-SGen (Power outputs: Hotend0, Hotend1, Bed, Fan)
198 #define BOARD_MKS_SGEN_L 1764 // MKS_SGEN_L (Power outputs: Hotend0, Hotend1, Bed, Fan)
199
200 //
201 // SAM3X8E ARM Cortex M3
202 //
203
204 #define BOARD_DUE3DOM 1411 // DUE3DOM for Arduino DUE
205 #define BOARD_DUE3DOM_MINI 1412 // DUE3DOM MINI for Arduino DUE
206 #define BOARD_RADD5 1502 // RADD5
207 #define BOARD_RAMPS_FD_V1 1503 // RAMPS-FD v1
208 #define BOARD_RAMPS_FD_V2 1504 // RAMPS-FD v2
209 #define BOARD_RAMPS_SMART_EFB 1523 // RAMPS-SMART (Power outputs: Hotend, Fan, Bed)
210 #define BOARD_RAMPS_SMART_EEB 1524 // RAMPS-SMART (Power outputs: Hotend0, Hotend1, Bed)
211 #define BOARD_RAMPS_SMART_EFF 1525 // RAMPS-SMART (Power outputs: Hotend, Fan0, Fan1)
212 #define BOARD_RAMPS_SMART_EEF 1526 // RAMPS-SMART (Power outputs: Hotend0, Hotend1, Fan)
213 #define BOARD_RAMPS_SMART_SF 1528 // RAMPS-SMART (Power outputs: Spindle, Controller Fan)
214 #define BOARD_RAMPS_DUO_EFB 1533 // RAMPS Duo (Power outputs: Hotend, Fan, Bed)
  
```

Configure the board type to “BOARD_MKS_SGEN_L” in the “configuration.h”



```

Configuration.h — D:\MarlinFirmware — Atom
File Edit View Selection Find Packages Help PlatformIO Auto Build

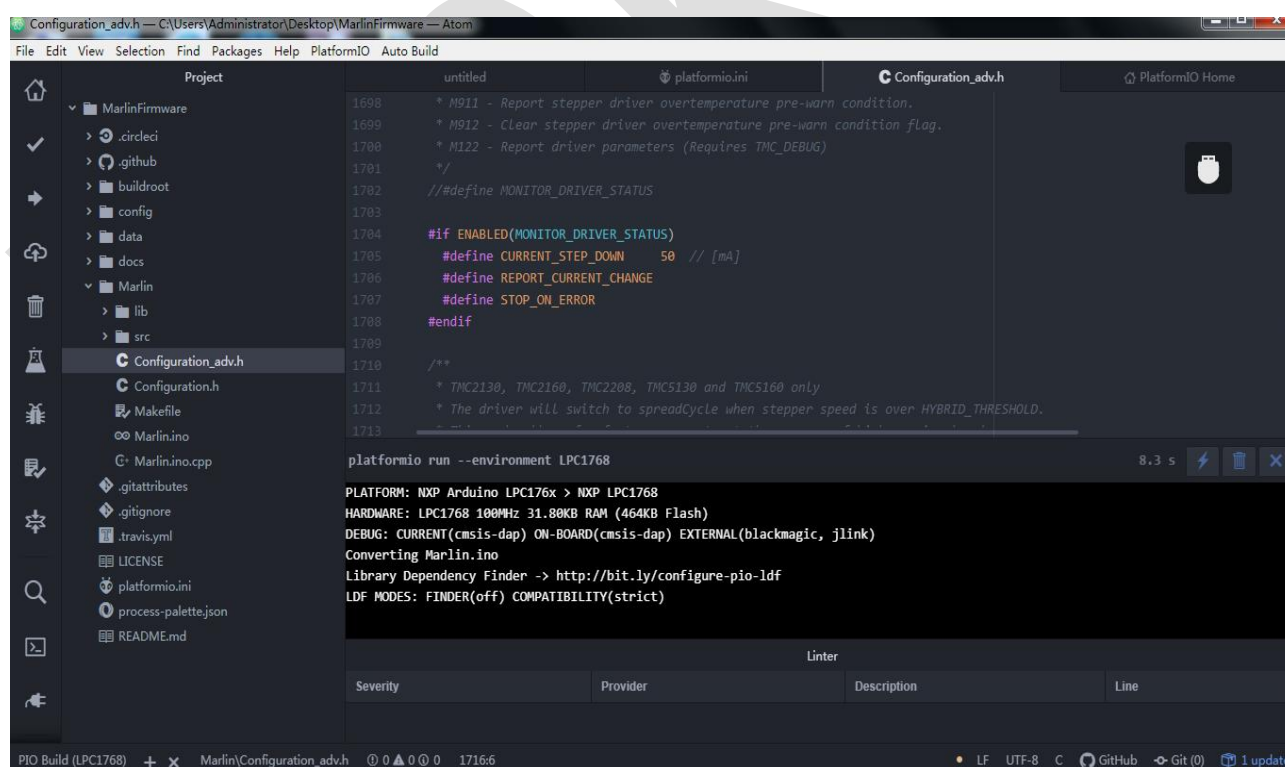
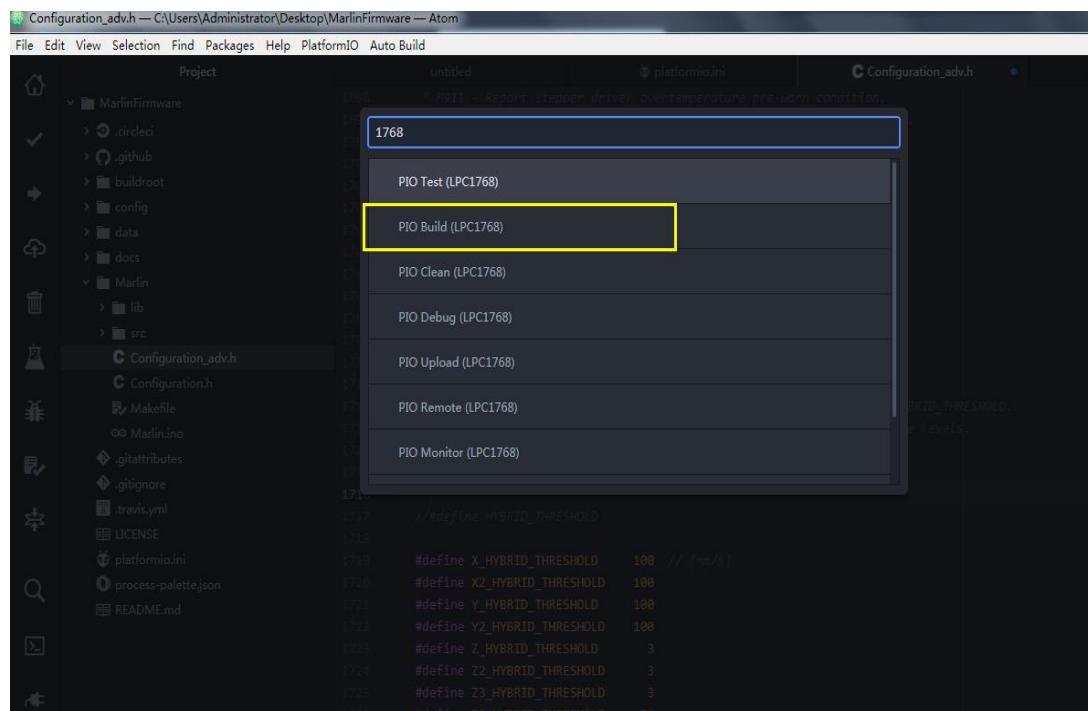
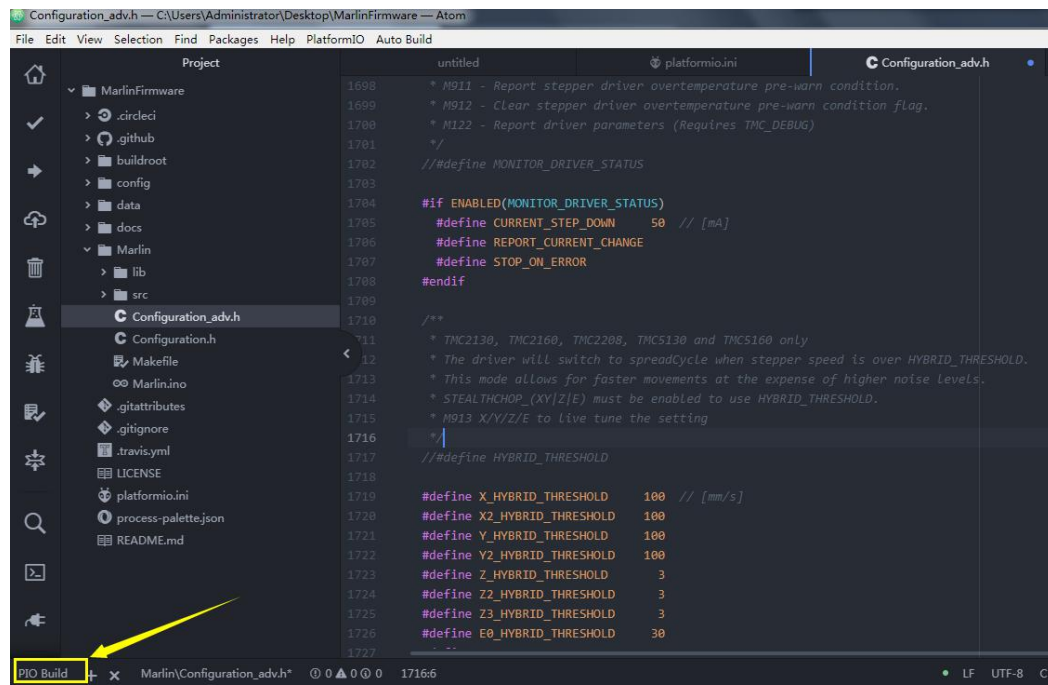
Project: MarlinFirmware
  .circleci
  .github
  buildroot
  config
  default
  examples
  data
  www
  docs
  Marlin
  lib
  src
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  .gitattributes
  .gitignore
  .travis.yml
  LICENSE
  platformio.ini
  process-palette.json
  README.md

Telemetry Consent Welcome Configuration.h
120 *
121 * 250000 works in most cases, but you might try a lower speed if
122 * you commonly experience drop-outs during host printing.
123 * You may try up to 1000000 to speed up SD file transfer.
124 *
125 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
126 */
127 #define BAUDRATE 115200
128
129 // Enable the Bluetooth serial interface on AT90USB devices
130 // #define BLUETOOTH
131
132 // The following define selects which electronics board you have.
133 // Please choose the name from boards.h that matches your setup
134 #ifndef MOTHERBOARD
135 #define MOTHERBOARD BOARD_MKS_SGEN_L // RAMPS_14_EFB
136 #endif
137
138 // Optional custom name for your RepStrap or other custom machine
139 // Displayed in the LCD "Ready" message
140 // #define CUSTOM_MACHINE_NAME "3D Printer"
141
142 // Define this to set a unique identifier for this printer, (Used by some programs to differentiate between machines)
143 // You can use an online service to generate a random UUID. (eg http://www.uuidgenerator.net)
144 // #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
145
  
```

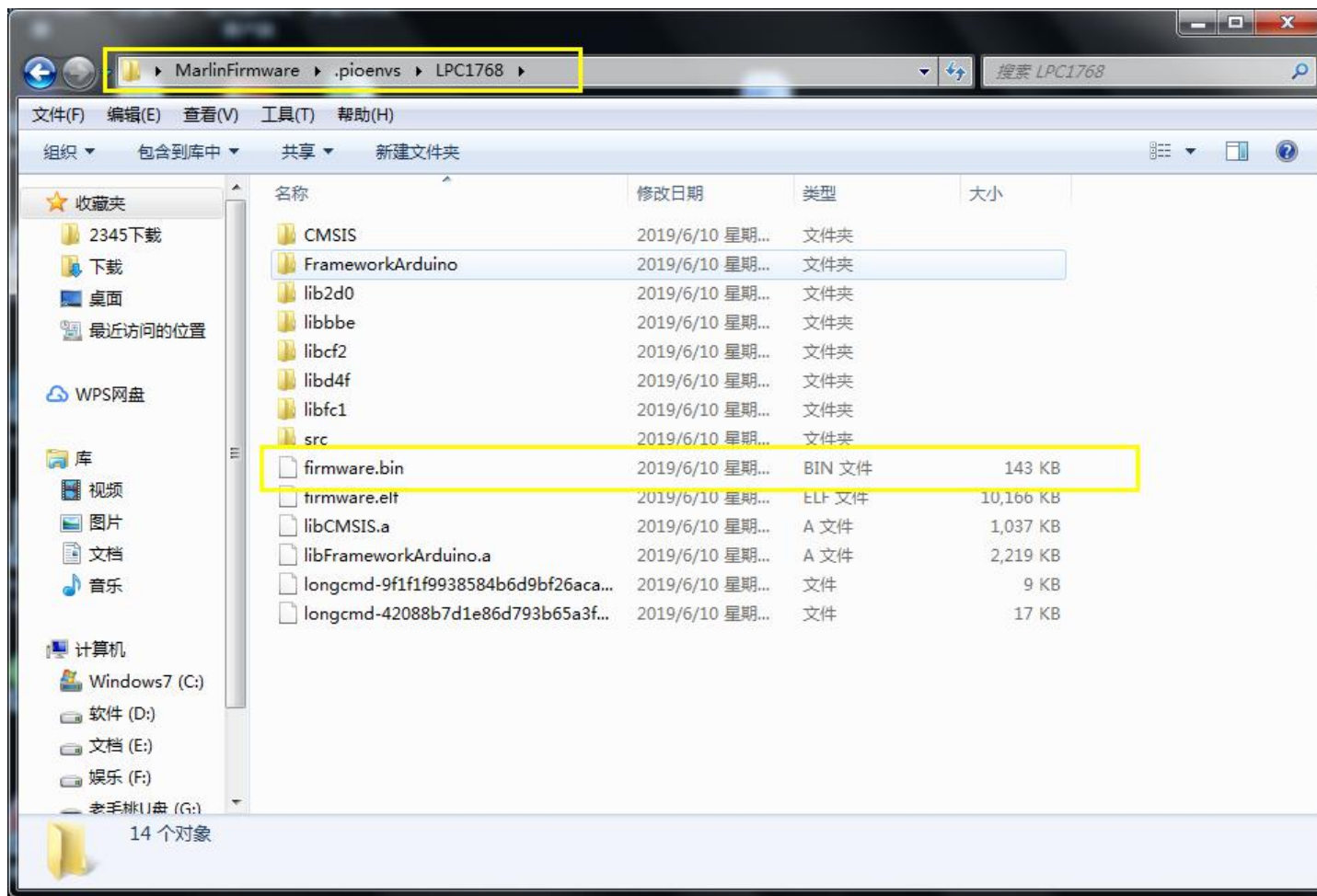
3. Setting parameters according different machine (refer to basic parameter configuration of marlin2.0 for details)

4. Firmware compilation

Click the PIO Build in the lower left corner→Input “176” →PIO Clean(LPC1768)→PIO Build(LPC1768). After compiling, open the marlin firmware folder→Open “.pioenvs”→Copy “firmware.bin”to TF card to insert into the motherboard’s card slot to flash firmware.



After the compilation is built, there will be a “firmware.bin” in this path, copy the file to TF card root directory to update.



V parameter configuration instruction

5.1 Smoothieware parameter configuration

1. Smoothieware basic parameter configuration

(1) The following parameter modification is based to different machines

Directly modify the “config.txt” file on the SD card, save it and power it on again. The parameter configuration takes effect at once.

```
#以下是经常需要的关键参数
alpha_steps_per_mm      80          # X轴每走1mm 所需脉冲数,
                          #例如 200步电机 16细分 用20-2GT同步轮: 200*16/40=80
beta_steps_per_mm       80          # Y轴每走1mm 所需脉冲数
gamma_steps_per_mm      1600       # Z轴每走1mm 所需脉冲数
extruder.hotend.steps_per_mm 140    # E0轴每走1mm 所需脉冲数

#alpha_current          1.0         # X轴电流大小 单位为安培
#beta_current           1.0         # Y轴电流大小
#gamma_current          1.0         # Z轴电流大小
#delta_current          1.0         # E0轴电流大小

alpha_dir_pin           2.3         # X轴转动方向, 在后面增加! 可让电机反转
beta_dir_pin            0.20        # Y轴转动方向, 在后面增加! 可让电机反转
gamma_dir_pin           2.11        # Z轴转动方向, 在后面增加! 可让电机反转
extruder.hotend.dir_pin 0.11        # E0轴转动方向, 在后面增加! 可让电机反转

network.ip_address      192.168.3.221 # IP地址
network.ip_mask         255.255.255.0 # 子网掩码
network.ip_gateway      192.168.3.1  # 网关
..
```

Note: Since the motor driver has changed to the direct-plug one, the driver current cannot be adjusted in the configuration file. It is adjusted by the knob on the direct-plug driver (refer to the following descriptions for driver current adjustment and precautions).

(2) The following parameter that may need to be modified

```
#以下是可能需要修改的参数

default_feed_rate       4000        # 默认速度 (mm/分钟) for G1/G2/G3 moves
default_seek_rate       4000        # 默认速度 (mm/分钟) for G0 moves
acceleration            500         # 加速度 mm/平方秒.
z_acceleration          60          # Z轴加速度
junction_deviation      0.02        # 类似Marlin的 "max_jerk"
                          # 数值越小, 电机运行效果越好, 但是速度越慢

x_axis_max_speed        10000       # X轴最大速度 mm/min
y_axis_max_speed        10000       # Y轴最大速度 mm/min
z_axis_max_speed        100         # Z轴最大速度 mm/min

alpha_max_rate          10000.0     # 要和x_axis_max_speed 一致
beta_max_rate           10000.0     # 要和y_axis_max_speed 一致
gamma_max_rate          100.0       # 要和z_axis_max_speed 一致

extruder.hotend.default_feed_rate 600    # E0默认速度 (mm/分钟)
extruder.hotend.acceleration      500    # E0轴加速度
extruder.hotend.max_speed         50     # mm/s

alpha_min_endstop       1.29^!     # X轴min限位开关, 在后面增加! 可设置常开还是常闭
beta_min_endstop        1.27^!     # Y轴min限位开关, 在后面增加! 可设置常开还是常闭
gamma_min_endstop       1.25^!     # Z轴min限位开关, 在后面增加! 可设置常开还是常闭
```

The speed and acceleration for each axis can be set according to personal needs.

If clicking on “homing” but with no response, Consumers can modify the switch type of the corresponding axis.

(3) extrusion “E0” parameter setting

```
#打印头E0设置

extruder.hotend.enable          true          # Whether to activate the extruder module at all. All
configuration is ignored if false
extruder.hotend.step_pin       2.13       # Pin for extruder step signal
extruder.hotend.en_pin         2.12       # Pin for extruder enable signal

temperature_control.hotend.enable true          #
temperature_control.hotend.thermistor_pin 0.23       #
temperature_control.hotend.heater_pin     2.7        #
temperature_control.hotend.thermistor     RRRF100K   #
temperature_control.hotend.set_m_code     104        #
temperature_control.hotend.set_and_wait_m_code 109      #
temperature_control.hotend.designator     T          #
```

Default setting

(4) double extruder parameter setting

If using double extrusions, it needs to delete the following “#” before the configuration.

```
# 双打印头设置，如果需要使用双打印头，需要去掉以下设置的注释

#extruder.hotend2.enable          true          #
#extruder.hotend2.steps_per_mm    90           # E1轴每走1mm 所需脉冲数
#extruder.hotend2.default_feed_rate 600          # E1默认速度 ( mm/分钟 )
#extruder.hotend2.acceleration    500          # E1轴加速度
#extruder.hotend2.max_speed       50             # mm/s
#extruder.hotend2.step_pin        0.1            #
#extruder.hotend2.dir_pin         0.0            # E1轴转动方向，在后面增加！可让电机反转
#extruder.hotend2.en_pin          0.10           #
#epsilon_current                  1.2            # E1电流大小

#temperature_control.hotend2.enable true          #
#temperature_control.hotend2.thermistor_pin 0.25       #
#temperature_control.hotend2.heater_pin     2.6        #
#temperature_control.hotend2.thermistor     RRRF100K   #
#temperature_control.hotend2.set_m_code     104        #
#temperature_control.hotend2.set_and_wait_m_code 109      #
#temperature_control.hotend2.designator     T1          #
```

It is recommended to use the above default settings if you are not familiar with corresponding pin and type information.

(5) Heated bed setting

```

#热床设置
temperature_control.bed.enable      true      #
temperature_control.bed.thermistor_pin 0.24     #
temperature_control.bed.heater_pin    2.5      # 2.5
temperature_control.bed.thermistor    RRRF100K # see http://smoothieware.org/temperaturecontrol#toc5
#temperature_control.bed.beta         3960     # or set the beta value

temperature_control.bed.set_m_code    140      #
temperature_control.bed.set_and_wait_m_code 190     #
temperature_control.bed.designator    B        #

#temperature_control.bed.bang_bang    false     # set to true to use bang bang control rather than PID
#temperature_control.bed.hysteresis   2.0      # set to the temperature in degrees C to use as hysteresis
# when using bang bang

```

The figure above shows how to enable heated bed, “true” can be replaced by “false” (ban heated bed) if consumers don’t need this function.

temperature_control.module_name.thermistor RRRF100K are 100K NTC. If using another sensor, it need to enter the official firmware website to search the its corresponding items name.

.PID setting

Solve the problem of excessive temperature fluctuation caused by firmware parameter.

First run the M303 command, for example:

```
M303 E0 S190
```

For the case where the printing head E0 is often warmed up to 190 degree, automatically run PID .

#System Run for about 8 cycles, showing the following information

```

T: 190.4/190.0 @0 0 7/8
T: 190.2/190.0 @0 0 7/8
Cycle 7:
Max: 190.8 Min: 184.3 high time: 48.2s low time: 7.5s
Averages over last 3 cycles: Max: 81.8c Min: 79.0c high :
ku: 17.7607
tu: 23.7929
Trying:
Kp: 10.7
Ki: 0.045
Kd: 32
PID Autotune Complete! The settings above have been loaded in

```

Input PID value to “config.txt”, or save the value by running M500 directly.

(6) Fan setting

```
# 风扇设置
switch.fan.enable           true           #
switch.fan.input_on_command M106        #
switch.fan.input_off_command M107        #
switch.fan.output_pin      2.4         #
switch.fan.output_type     pwm          # pwm output settable with S parameter in the
input_on_comand
```

(7) Limit switch setting

```
#限位开关设置
endstops_enable            true           # the endstop module is enabled by default and can be
disabled here

alpha_max_endstop         1.28^         #
alpha_homing_direction    home_to_min  # or set to home_to_max and set alpha_max
alpha_min                 0             # this gets loaded after homing when home_to_min is set
alpha_max                 250          # this gets loaded after homing when home_to_max is set

beta_max_endstop          1.26^         #
beta_homing_direction     home_to_min  #
beta_min                 0             #
beta_max                 250          #

gamma_max_endstop         1.24^         #
gamma_homing_direction    home_to_min  #
gamma_min                 0             #
gamma_max                 120         #

alpha_fast_homing_rate_mm_s 50         # feedrates in mm/second
beta_fast_homing_rate_mm_s 50         # "
gamma_fast_homing_rate_mm_s 4          # "
alpha_slow_homing_rate_mm_s 25        # "
beta_slow_homing_rate_mm_s 25         # "
gamma_slow_homing_rate_mm_s 2         # "

alpha_homing_retract_mm   5           # distance in mm
beta_homing_retract_mm   5           # "
gamma_homing_retract_mm  1           # "
```

If you want to use the limit switch, "endstops_enable" must be set to "true".refer to the following figure.

Alpha, beta, and gamma correspond to three axes.

Homing_direction: direction of homing, "home_to_min": minimum, "home_to_max": maximum.

"Alpha_min": minimum, "alpha_max" maximum. The other axes are the same as described above.

(8) Auto-leveling setting

```
# 自动调平设置
zprobe.enable             false        # set to true to enable a zprobe
zprobe.probe_pin          1.25!^      # pin probe is attached to if NC remove the !
zprobe.slow_feedrate      5           # mm/sec probe feed rate
#zprobe.debounce_count    100         # set if noisy
zprobe.fast_feedrate      100         # move feedrate mm/sec
zprobe.probe_height       5           # how much above bed to start probe
```

Only suitable for ordinal switch setting

In leveling process, using the PC software to perform the following process:

1) G32

#perform the following process

2) G28

#homing

3) G0 Z5

#move Z axis up 5mm

4) Manually adjust height of Z-axis until the distance between the printhead and the heated bed is as one paper thickness.

5) M306 Z0

#configure the height to "0"

6) G28

#homing again

7) G0 Z1

#Set the current height to 1mm, and measure whether the printing head is 1mm away from the heat bed.

8) M500

#save current data to EEPROM;

#Note: After executing M500, the system will not read the parameters from config.txt. After running M502 to clear the parameters, the parameters will be read from config.txt at the next startup.

(9) LCD display setting

MKS LCD MINI12864/MKS LCD12864A (12864A needs to remove RPK2 resistor on PCB)

Screen display settings (add # before the line for shielding, or delete the line, the screen will work properly, refer to the following picture)


```

# MKS LCD12864/MINI12864控制面板设置
panel.enable          true          # set to true to enable the panel code
panel.lcd             st7565_glcd    # MKS MINI12864
panel.spi_channel     1             # spi channel to use ; GLCD EXP1 Pins 3,5 (MOSI, SCLK)
panel.spi_cs_pin      0.17          # spi chip select ; GLCD EXP1 Pin 4
panel.spi_frequency   500000        # SPI port frequency - some panel need it explicitly set
panel.contrast        0             # Contrast value for panels that support it
panel.encoder_resolution 4
panel.reverse         true          # If set to true, reverse the screen.
panel.busy_pin        nc            #
panel.a0_pin          1.0           # spi A0
panel.rst_pin         nc            #
panel.menu_offset     0             #

panel.encoder_a_pin   3.25!^        # encoder pin ; GLCD EXP2 Pin 3
panel.encoder_b_pin   3.26!^        # encoder pin ; GLCD EXP2 Pin 5
panel.click_button_pin 1.30!^       # click button ; GLCD EXP1 Pin 2
panel.buzz_pin        1.31          # pin for buzzer ; GLCD EXP1 Pin 1
#panel.back_button_pin 2.11!^       # back button ; GLCD EXP2 Pin 8

panel.external_sd     true          # set to true if there is an external sdcard on the panel
panel.external_sd.spi_channel 1      # set spi channel the sdcard is on
panel.external_sd.spi_cs_pin 0.28    # set spi chip select for the sdcard (or any spare pin)
panel.external_sd.sdcd_pin 0.27!^    # sd detect signal (set to nc if no sdcard detect) (or any spare pin)

panel.menu_offset     1             # some panels will need 1 here
panel.alpha_jog_feedrate 6000      # x jogging feedrate in mm/min
panel.beta_jog_feedrate 6000       # y jogging feedrate in mm/min
panel.gamma_jog_feedrate 200       # z jogging feedrate in mm/min
panel.hotend_temperature 185       # temp to set hotend when preheat is selected
panel.bed_temperature 60           # temp to set bed when preheat is selected

```

MKS LCD 12864 screen setting

You need to add the “#” line (as the red high-lighted line shows) to make the screen operate normally

```

# MKS 12864控制面板设置
panel.enable          true          # set to true to enable the panel code
panel.lcd             reprap_discount_glcd #
panel.spi_channel     0             # spi channel to use ; GLCD EXP1 Pins 3,5 (MOSI, SCLK)
panel.spi_cs_pin      0.16          # spi chip select ; GLCD EXP1 Pin 4
panel.encoder_a_pin   3.25!^        # encoder pin ; GLCD EXP2 Pin 3
panel.encoder_b_pin   3.26!^        # encoder pin ; GLCD EXP2 Pin 5
panel.click_button_pin 1.30!^       # click button ; GLCD EXP1 Pin 2
panel.buzz_pin        1.31          # pin for buzzer ; GLCD EXP1 Pin 1
#panel.back_button_pin 2.11!^       # back button ; GLCD EXP2 Pin 8

panel.external_sd     true          # set to true if there is an external sdcard on the panel
panel.external_sd.spi_channel 1      # set spi channel the sdcard is on
panel.external_sd.spi_cs_pin 0.28    # set spi chip select for the sdcard (or any spare pin)
panel.external_sd.sdcd_pin 0.27!^    # sd detect signal (set to nc if no sdcard detect) (or any spare pin)

panel.menu_offset     1             # some panels will need 1 here
panel.alpha_jog_feedrate 6000      # x jogging feedrate in mm/min
panel.beta_jog_feedrate 6000       # y jogging feedrate in mm/min
panel.gamma_jog_feedrate 200       # z jogging feedrate in mm/min
panel.hotend_temperature 185       # temp to set hotend when preheat is selected
panel.bed_temperature 60           # temp to set bed when preheat is selected

```

5.2 marlin2.0 parameter setting

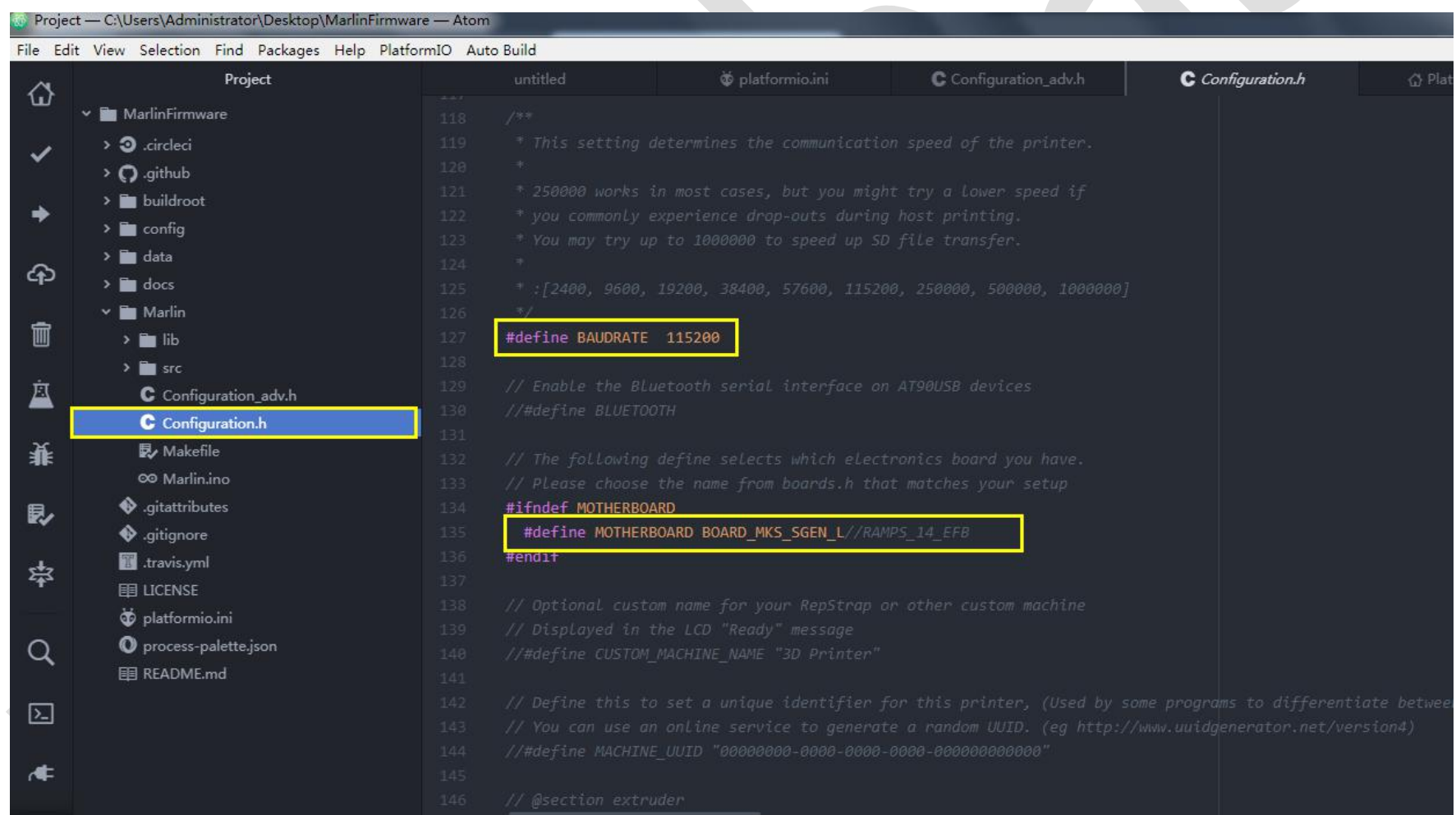
5.2.1 basic parameter setting

Marlin2.0 parameter can be modified and built directly via “Atom” software.

Open marlin firmware via atom, click on “File→Open Folder” (open marlin2.0wirmware folder to input marlin firmware. Open “configuration.h” to setting the basic parameter. The setting contact is similar to marlin version 1.0.

1. Motherboard Baud rate setting

2. motherboard type setting



```
Project — C:\Users\Administrator\Desktop\MarlinFirmware — Atom
File Edit View Selection Find Packages Help PlatformIO Auto Build
Project
  MarlinFirmware
    .circleci
    .github
    buildroot
    config
    data
    docs
    Marlin
      lib
      src
      Configuration_adv.h
      Configuration.h
      Makefile
      Marlin.ino
      .gitattributes
      .gitignore
      .travis.yml
      LICENSE
      platformio.ini
      process-palette.json
      README.md
untitled
platformio.ini
Configuration_adv.h
Configuration.h
118  /**
119  * This setting determines the communication speed of the printer.
120  *
121  * 250000 works in most cases, but you might try a lower speed if
122  * you commonly experience drop-outs during host printing.
123  * You may try up to 1000000 to speed up SD file transfer.
124  *
125  *:[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
126  */
127  #define BAUDRATE 115200
128
129  // Enable the Bluetooth serial interface on AT90USB devices
130  //#define BLUETOOTH
131
132  // The following define selects which electronics board you have.
133  // Please choose the name from boards.h that matches your setup
134  #ifndef MOTHERBOARD
135  #define MOTHERBOARD BOARD_MKS_SGEN_L//RAMPS_14_EFB
136  #endif
137
138  // Optional custom name for your RepStrap or other custom machine
139  // Displayed in the LCD "Ready" message
140  //#define CUSTOM_MACHINE_NAME "3D Printer"
141
142  // Define this to set a unique identifier for this printer, (Used by some programs to differentiate between
143  // You can use an online service to generate a random UUID. (eg http://www.uuidgenerator.net/version4)
144  //#define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
145
146  // @section extruder
```


Extruder quantity and thermal type setting

```

138 // Optional custom name for your RepStrap or other custom machine
139 // Displayed in the LCD "Ready" message
140 // #define CUSTOM_MACHINE_NAME "3D Printer"
141
142 // Define this to set a unique identifier for this printer, (Used by some programs to differentiate
143 // You can use an online service to generate a random UUID. (eg http://www.uuidgenerator.net/vers
144 // #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
145
146 // @section extruder
147
148 // This defines the number of extruders
149 // {1, 2, 3, 4, 5, 6}
150 #define EXTRUDERS 1
151
152 // Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used for Volumetric, Filament Wid
153 #define DEFAULT_NOMINAL_FILAMENT_DIA 3.0
154
155 // For Cyclops or any "multi-extruder" that shares a single nozzle.
156 // #define SINGLENOZZLE
157
158 /**
159 * Průša MK2 Single Nozzle Multi-Material Multiplexer, and variants.
160 *
161 * This device allows one stepper driver on a control board to drive
162 * two to eight stepper motors, one at a time, in a manner suitable
163 * for extruders.
164 *
165 * This option only allows the multiplexer to switch on tool-change.
166 * Additional options to configure custom E moves are pending.
167 */
    
```

```

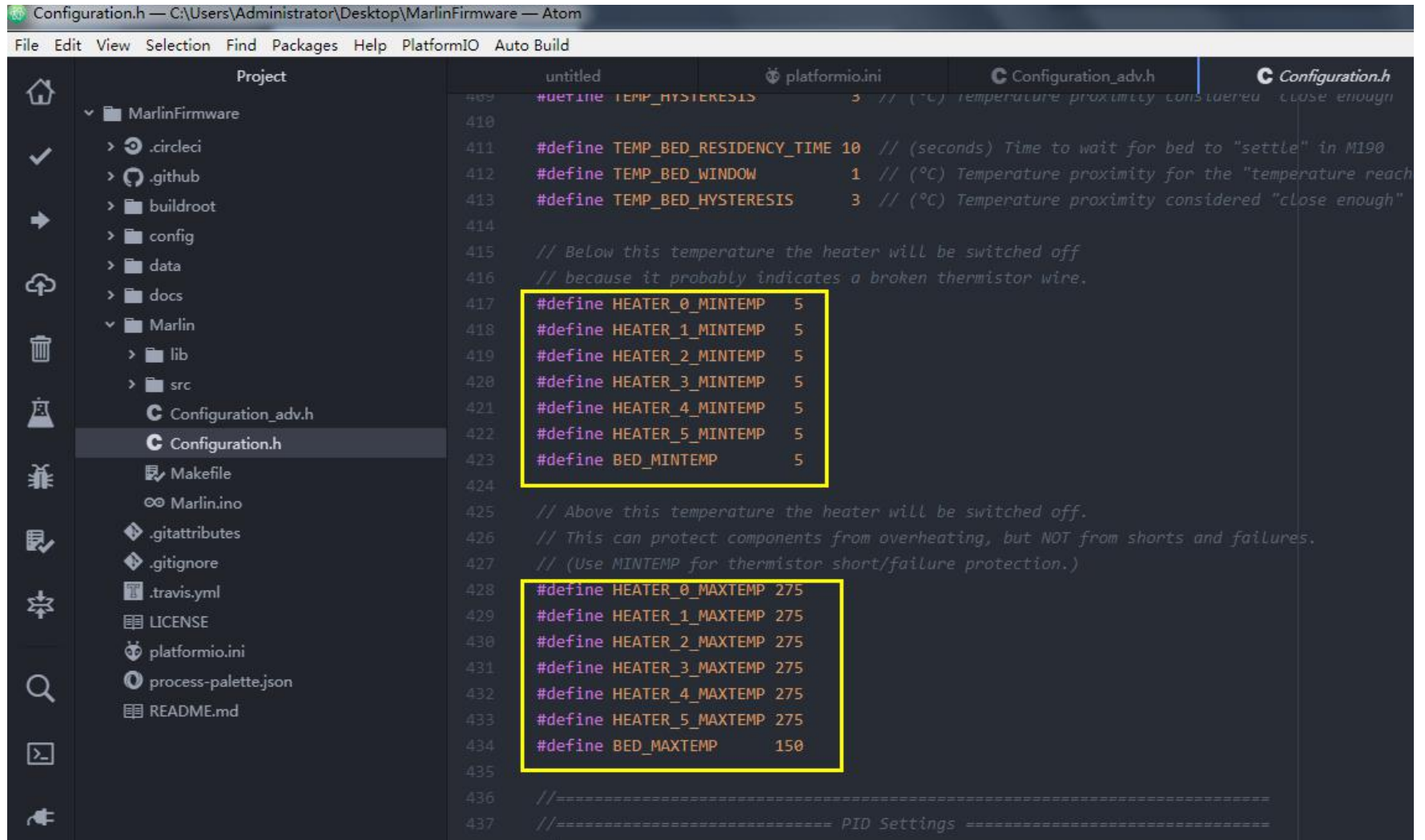
374 * 55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)
375 *
376 * 1047 : Pt1000 with 4k7 pullup
377 * 1010 : Pt1000 with 1k pullup (non standard)
378 * 147 : Pt100 with 4k7 pullup
379 * 110 : Pt100 with 1k pullup (non standard)
380 *
381 * 1000 : Custom - Specify parameters in Configuration_adv.h
382 *
383 * Use these for Testing or Development purposes. NEVER for production machine.
384 * 998 : Dummy Table that ALWAYS reads 25°C or the temperature defined below.
385 * 999 : Dummy Table that ALWAYS reads 100°C or the temperature defined below.
386 *
387 * :{ '0':"Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC Semitec 204GT-2", '3':
388 * '10':"100k / 4.7k RS 198-961", '11':"100k / 4.7k beta 3950 1%", '12':"100k / 4.7k 0603 SMD VI
389 * 3950 350C thermistor 4.7k pullUp", '66':"Dyze Design 4.7M High Temperature thermistor", '67':
390 * '-1':"Thermocouple + AD595", '998':"Dummy 1", '999':"Dummy 2", '1000':"Custom thermistor para
391
392 #define TEMP_SENSOR_0 1
393 #define TEMP_SENSOR_1 0
394 #define TEMP_SENSOR_2 0
395 #define TEMP_SENSOR_3 0
396 #define TEMP_SENSOR_4 0
397 #define TEMP_SENSOR_5 0
398 #define TEMP_SENSOR_BED 1
399 #define TEMP_SENSOR_CHAMBER 0
400
401 // Dummy thermistor constant temperature readings, for use with 998 and 999
402 #define DUMMY_THERMISTOR_998_VALUE 25
    
```

Configure parameter according to the type of temperature sensor

#define TEMP_SENSOR_BED 1 (this item is the heated bed setting item, set to “0” means heated bed is not enabled, and other numbers are corresponding to sensors)

Temperature setting

Maximum and minimum temperature setting (if detected the actual temperature is not within this range, the motherboard will report an error automatically)



```

409 #define TEMP_HYSTERESIS 3 // (°C) temperature proximity considered "close enough"
410
411 #define TEMP_BED_RESIDENCY_TIME 10 // (seconds) Time to wait for bed to "settle" in M190
412 #define TEMP_BED_WINDOW 1 // (°C) Temperature proximity for the "temperature reach
413 #define TEMP_BED_HYSTERESIS 3 // (°C) Temperature proximity considered "close enough"
414
415 // Below this temperature the heater will be switched off
416 // because it probably indicates a broken thermistor wire.
417 #define HEATER_0_MINTEMP 5
418 #define HEATER_1_MINTEMP 5
419 #define HEATER_2_MINTEMP 5
420 #define HEATER_3_MINTEMP 5
421 #define HEATER_4_MINTEMP 5
422 #define HEATER_5_MINTEMP 5
423 #define BED_MINTEMP 5
424
425 // Above this temperature the heater will be switched off.
426 // This can protect components from overheating, but NOT from shorts and failures.
427 // (Use MINTEMP for thermistor short/failure protection.)
428 #define HEATER_0_MAXTEMP 275
429 #define HEATER_1_MAXTEMP 275
430 #define HEATER_2_MAXTEMP 275
431 #define HEATER_3_MAXTEMP 275
432 #define HEATER_4_MAXTEMP 275
433 #define HEATER_5_MAXTEMP 275
434 #define BED_MAXTEMP 150
435
436 //=====
437 //===== PID Settings =====

```

PID setting (generally, it doesn't need to adjust, default setting is OK)

The adjustment method is the same as smoothieware. Use the M303 command, for example:

```
M303 E0 S190
```

For the case where the printing head E0 is often warmed to 190 degrees, the PID operates automatically.

After that, fill the returned value into the firmware.

```

T: 190.4/190.0 @0 0 7/8
T: 190.2/190.0 @0 0 7/8
Cycle 7:
Max: 190.8 Min: 184.3 high time: 48.2s low time: 7.5s
Averages over last 3 cycles: Max: 81.8c Min: 79.0c high :
ku: 17.7607
tu: 23.7929
Trying:
Kp: 10.7
Ki: 0.045
Kd: 32
PID Autotune Complete! The settings above have been loaded in

```

```

451 // #define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.125Hz=8s) and mi
452 // #define PID_PARAMS_PER_HOTEND // Uses separate PID parameters for each extruder (useful
453 // Set/get with gcode: M301 E[extruder number, 0-2]
454 #define PID_FUNCTIONAL_RANGE 10 // If the temperature difference between the target temper
455 // is more than PID_FUNCTIONAL_RANGE then the PID will be
456
457 // If you are using a pre-configured hotend then you can use one of the value sets by unco
458
459 // Ultimaker
460 // #define DEFAULT_Kp 22.2
461 // #define DEFAULT_Ki 1.08
462 // #define DEFAULT_Kd 114
463
464 #define DEFAULT_Kp 12.71
465 #define DEFAULT_Ki 0.61
466 #define DEFAULT_Kd 65.92
467 // MakerGear
468 // #define DEFAULT_Kp 7.0
469 // #define DEFAULT_Ki 0.1
470 // #define DEFAULT_Kd 12
471
472 // Mendel Parts V9 on 12V
473 // #define DEFAULT_Kp 63.0
474 // #define DEFAULT_Ki 2.25
475 // #define DEFAULT_Kd 440
476
477 #endif // PIDTEMP
478
479 //

```

PREVENT_COLD_EXTRUSION

It is 170 degree by default, only when warmed up to 170 degree can the extruder work. You need to reduce the extruder temperature if it doesn't need to warm up.

```

526
527 // @section extruder
528
529 /**
530  * Prevent extrusion if the temperature is below EXTRUDE_MINTEMP.
531  * Add M302 to set the minimum extrusion temperature and/or turn
532  * cold extrusion prevention on and off.
533  *
534  * *** IT IS HIGHLY RECOMMENDED TO LEAVE THIS OPTION ENABLED! ***
535  */
536 #define PREVENT_COLD_EXTRUSION
537 #define EXTRUDE_MINTEMP 170
538
539 /**
540  * Prevent a single extrusion longer than EXTRUDE_MAXLENGTH.
541  * Note: For Bowden Extruders make this large enough to allow load/unload.
542  */
543 #define PREVENT_LENGTHY_EXTRUDE
544 #define EXTRUDE_MAXLENGTH 200
545
546 //=====
547 //===== Thermal Runaway Protection =====
548 //=====
549
550 /**
551  * Thermal Protection provides additional protection to your printer from damage
552  * and fire. Marlin always includes safe min and max temperature ranges which
553  * protect against a broken or disconnected thermistor wire.
554  *
555  *

```


Machine setting

Machine structure setting, default structure is xyz. If it is corexy or other structure, it needs to enabled the corresponding type.

```

563 #define THERMAL_PROTECTION_HOTENDS // Enable thermal protection for all extruders
564 #define THERMAL_PROTECTION_BED // Enable thermal protection for the heated bed
565 #define THERMAL_PROTECTION_CHAMBER // Enable thermal protection for the heated chamber
566
567 //=====
568 //===== Mechanical Settings =====
569 //=====
570
571 // @section machine
572
573 // Uncomment one of these options to enable CoreXY, CoreXZ, or CoreYZ kinematics
574 // either in the usual order or reversed
575 // #define COREXY
576 // #define COREXZ
577 // #define COREYZ
578 // #define COREYX
579 // #define COREZX
580 // #define COREZY
581
582 //=====
583 //===== Endstop Settings =====
584 //=====
585
586 // @section homing
587
588 // Specify here all the endstop connectors that are connected to any endstop or probe.
589 // Almost all printers will be using one per axis. Probes will use one or more of the
590 // extra connectors. Leave undefined any used for non-endstop and non-probe purposes.
591 #define USE_XMIN_PLUG
    
```

Limit switch setting

Enable limit switch

```

579 // #define COREXZ
580 // #define COREZY
581
582 //=====
583 //===== Endstop Settings =====
584 //=====
585
586 // @section homing
587
588 // Specify here all the endstop connectors that are connected to any endstop or probe.
589 // Almost all printers will be using one per axis. Probes will use one or more of the
590 // extra connectors. Leave undefined any used for non-endstop and non-probe purposes.
591 #define USE_XMIN_PLUG
592 #define USE_YMIN_PLUG
593 #define USE_ZMIN_PLUG
594 // #define USE_XMAX_PLUG
595 // #define USE_YMAX_PLUG
596 // #define USE_ZMAX_PLUG
597
598 // Enable pullup for all endstops to prevent a floating state
599 // #define ENDSTOPPULLUPS
600 #if DISABLED(ENDSTOPPULLUPS)
601 // Disable ENDSTOPPULLUPS to set pullups individually
602 #define ENDSTOPPULLUP_XMAX
603 #define ENDSTOPPULLUP_YMAX
604 #define ENDSTOPPULLUP_ZMAX
605 #define ENDSTOPPULLUP_XMIN
606 #define ENDSTOPPULLUP_YMIN
607 #define ENDSTOPPULLUP_ZMIN
    
```


Switch type setting (normally open / normally closed)

Note: If the setting type is reversed, it may cause homing error.

```

613  #if DISABLED(ENDSTOPPULLDOWNS)
614  // Disable ENDSTOPPULLDOWNS to set pulldowns individually
615  // #define ENDSTOPPULLDOWN_XMAX
616  // #define ENDSTOPPULLDOWN_YMAX
617  // #define ENDSTOPPULLDOWN_ZMAX
618  // #define ENDSTOPPULLDOWN_XMIN
619  // #define ENDSTOPPULLDOWN_YMIN
620  // #define ENDSTOPPULLDOWN_ZMIN
621  // #define ENDSTOPPULLDOWN_ZMIN_PROBE
622  #endif
623
624  // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
625  #define X_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
626  #define Y_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
627  #define Z_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
628  #define X_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
629  #define Y_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
630  #define Z_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
631  #define Z_MIN_PROBE_ENDSTOP_INVERTING false // set to true to invert the logic of the probe.
632
633  /**
634   * Stepper Drivers
635   *
636   * These settings allow Marlin to tune stepper driver timing and enable advanced options for
637   * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
638   *
639   * A4988 is assumed for unspecified drivers.
640   *
641   * Options: A4988, A5984, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
  
```

Driver type

Only when driver type is confirmed, can it configure the corresponding special mode (TMC2208 is compatible with Uart mode, TMC2130 compatible with SPI mode)

The driver is set to normal mode by default without enabling corresponding item.

```

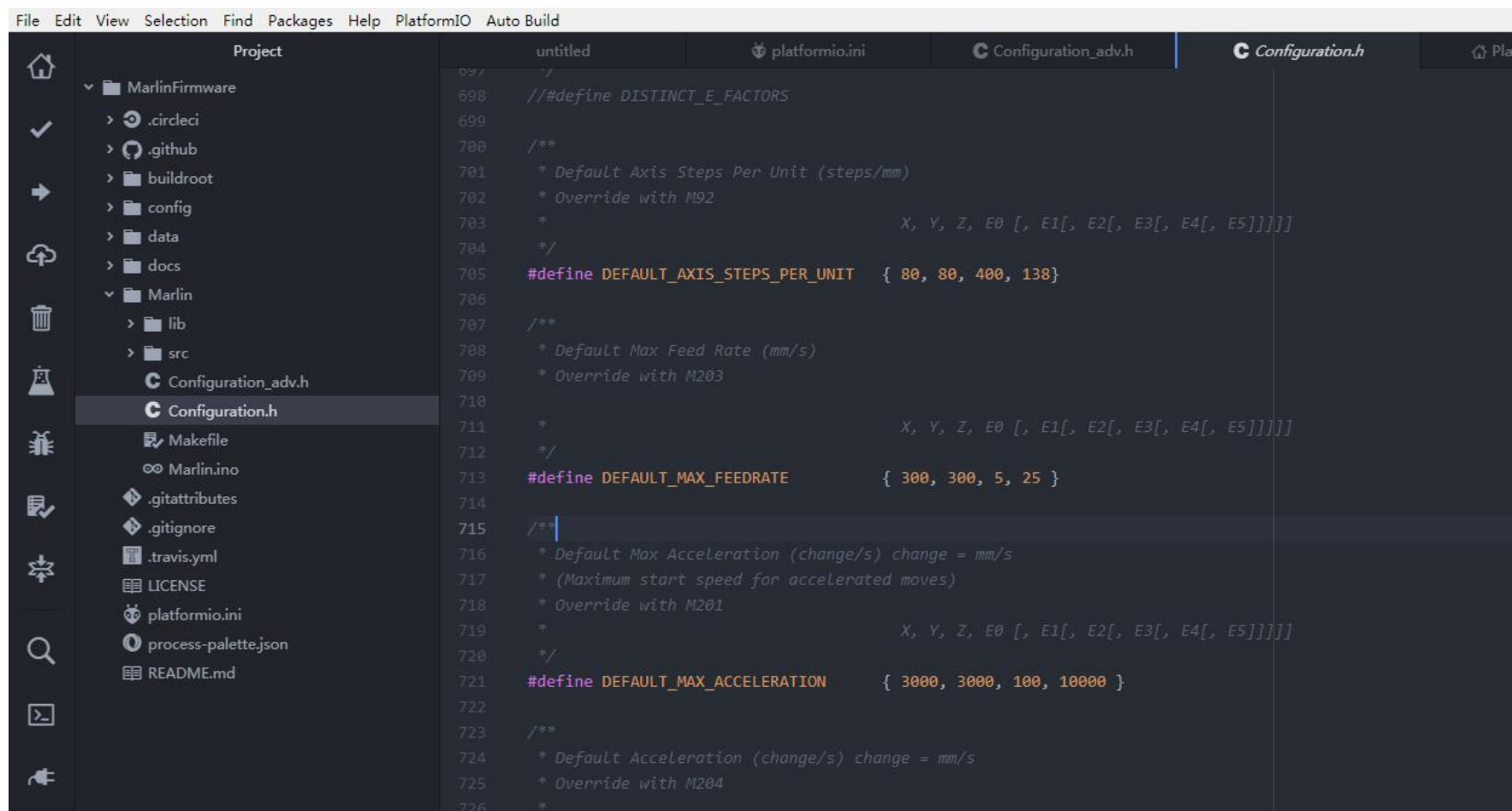
636  * These settings allow Marlin to tune stepper driver timing and enable advanced options for
637  * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
638  *
639  * A4988 is assumed for unspecified drivers.
640  *
641  * Options: A4988, A5984, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
642  *          TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
643  *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
644  *          TMC2160, TMC2160_STANDALONE, TMC5130, TMC5130_STANDALONE,
645  *          TMC5160, TMC5160_STANDALONE
646  * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC21
647  */
648  #define X_DRIVER_TYPE  TMC2208
649  #define Y_DRIVER_TYPE  TMC2208
650  #define Z_DRIVER_TYPE  TMC2208
651  // #define X2_DRIVER_TYPE  A4988
652  // #define Y2_DRIVER_TYPE  A4988
653  // #define Z2_DRIVER_TYPE  TMC2130
654  // #define Z3_DRIVER_TYPE  A4988
655  #define E0_DRIVER_TYPE  TMC2208
656  // #define E1_DRIVER_TYPE  A4988
657  // #define E2_DRIVER_TYPE  A4988
658  // #define E3_DRIVER_TYPE  A4988
659  // #define E4_DRIVER_TYPE  A4988
660  // #define E5_DRIVER_TYPE  A4988
661
662  // Enable this feature if all enabled endstop pins are interrupt-capable.
663  // This will remove the need to poll the interrupt pins, saving many CPU cycles.
664  // #define ENDSTOP_INTERRUPTS_FEATURE
  
```

Motor steps per setting (#define DEFAULT_AXIS_STEPS_PER_UNIT)

It can be calculated by formula according to mechanical conditions

The formula of the number of pulses of the synchronous wheel motor / mm is: $(360 \div \text{step angle}) \times \text{microstep}$
(diameter $\times 3.14$);

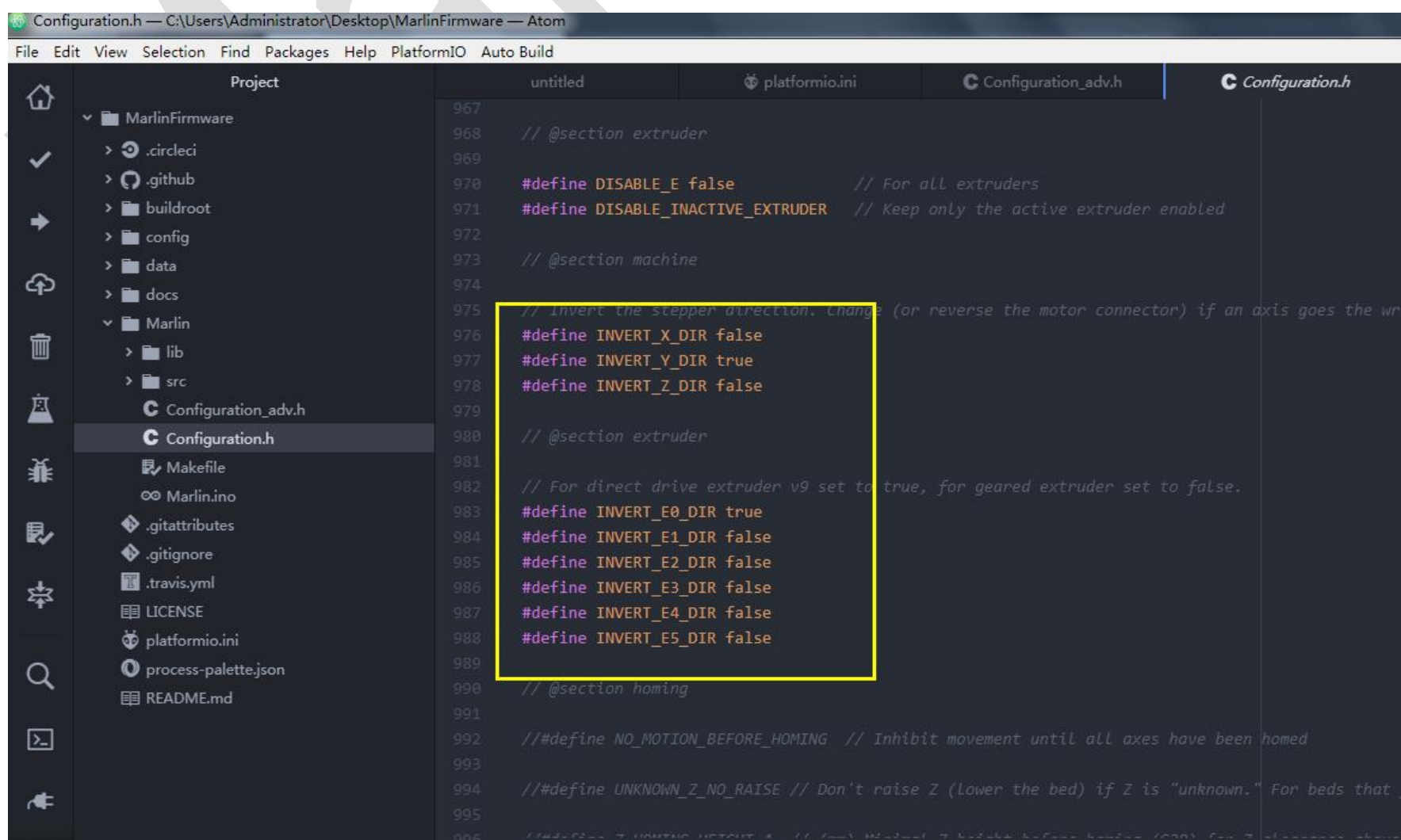
The formula of the number of pulses of the screw motor / mm is: $(360 \div \text{step angle}) \times \text{microstep} \div \text{lead}$



```
File Edit View Selection Find Packages Help PlatformIO Auto Build
Project
  MarlinFirmware
    .circleci
    .github
    buildroot
    config
    data
    docs
    Marlin
      lib
      src
      Configuration_adv.h
      Configuration.h
      Makefile
      Marlin.ino
      .gitattributes
      .gitignore
      .travis.yml
      LICENSE
      platformio.ini
      process-palette.json
      README.md
untitled
platformio.ini
Configuration_adv.h
Configuration.h
997 //
698 // #define DISTINCT_E_FACTORS
699
700 /**
701  * Default Axis Steps Per Unit (steps/mm)
702  * Override with M92
703  *
704  * X, Y, Z, E0 [, E1[, E2[, E3[, E4[, E5]]]]]
705  */
706 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 138 }
707
708 /**
709  * Default Max Feed Rate (mm/s)
710  * Override with M203
711  *
712  * X, Y, Z, E0 [, E1[, E2[, E3[, E4[, E5]]]]]
713  */
714 #define DEFAULT_MAX_FEEDRATE { 300, 300, 5, 25 }
715
716 /**
717  * Default Max Acceleration (change/s) change = mm/s
718  * (Maximum start speed for accelerated moves)
719  * Override with M201
720  *
721  * X, Y, Z, E0 [, E1[, E2[, E3[, E4[, E5]]]]]
722  */
723 #define DEFAULT_MAX_ACCELERATION { 3000, 3000, 100, 10000 }
724
725 /**
726  * Default Acceleration (change/s) change = mm/s
727  * Override with M204
728  *
729  * X, Y, Z, E0 [, E1[, E2[, E3[, E4[, E5]]]]]
730  */
731 #define DEFAULT_ACCELERATION { 5000, 5000, 100, 10000 }
```

Motor direction setting

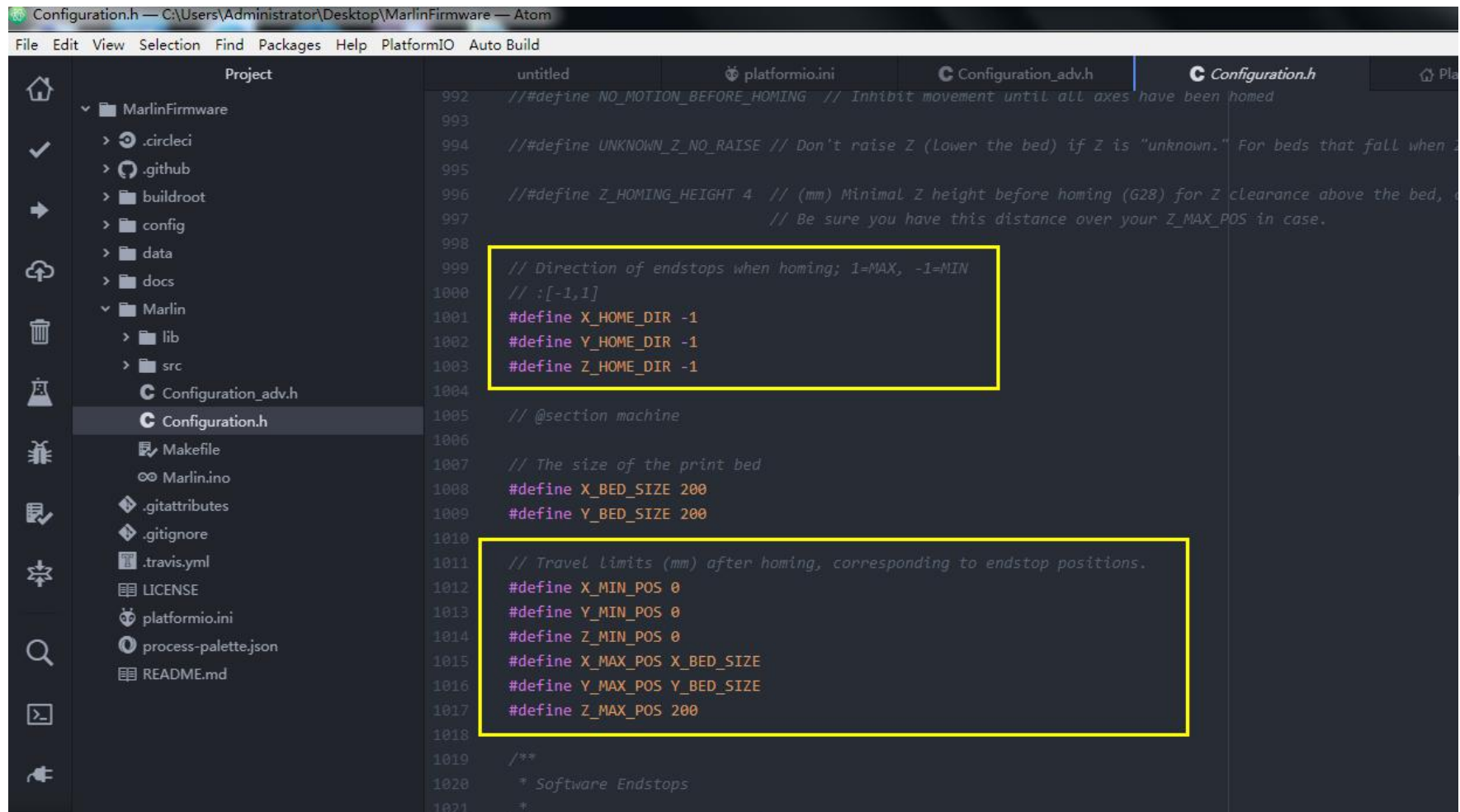
adjust parameter(true or false) to modified motor direction



```
Configuration.h — C:\Users\Administrator\Desktop\MarlinFirmware — Atom
File Edit View Selection Find Packages Help PlatformIO Auto Build
Project
  MarlinFirmware
    .circleci
    .github
    buildroot
    config
    data
    docs
    Marlin
      lib
      src
      Configuration_adv.h
      Configuration.h
      Makefile
      Marlin.ino
      .gitattributes
      .gitignore
      .travis.yml
      LICENSE
      platformio.ini
      process-palette.json
      README.md
untitled
platformio.ini
Configuration_adv.h
Configuration.h
967 //
968 // @section extruder
969
970 #define DISABLE_E false // For all extruders
971 #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
972
973 // @section machine
974
975 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
976 #define INVERT_X_DIR false
977 #define INVERT_Y_DIR true
978 #define INVERT_Z_DIR false
979
980 // @section extruder
981
982 // For direct drive extruder v9 set to true, for geared extruder set to false.
983 #define INVERT_E0_DIR true
984 #define INVERT_E1_DIR false
985 #define INVERT_E2_DIR false
986 #define INVERT_E3_DIR false
987 #define INVERT_E4_DIR false
988 #define INVERT_E5_DIR false
989
990 // @section homing
991
992 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
993
994 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that
995
996 // #define Z_HOME_FROM_MINUTE // (Z) M204 T [steps/mm] [steps/mm] [steps/mm] [steps/mm] [steps/mm] [steps/mm]
```


Direction of Homing: “-1” is the minimum, and “1” is the maximum.

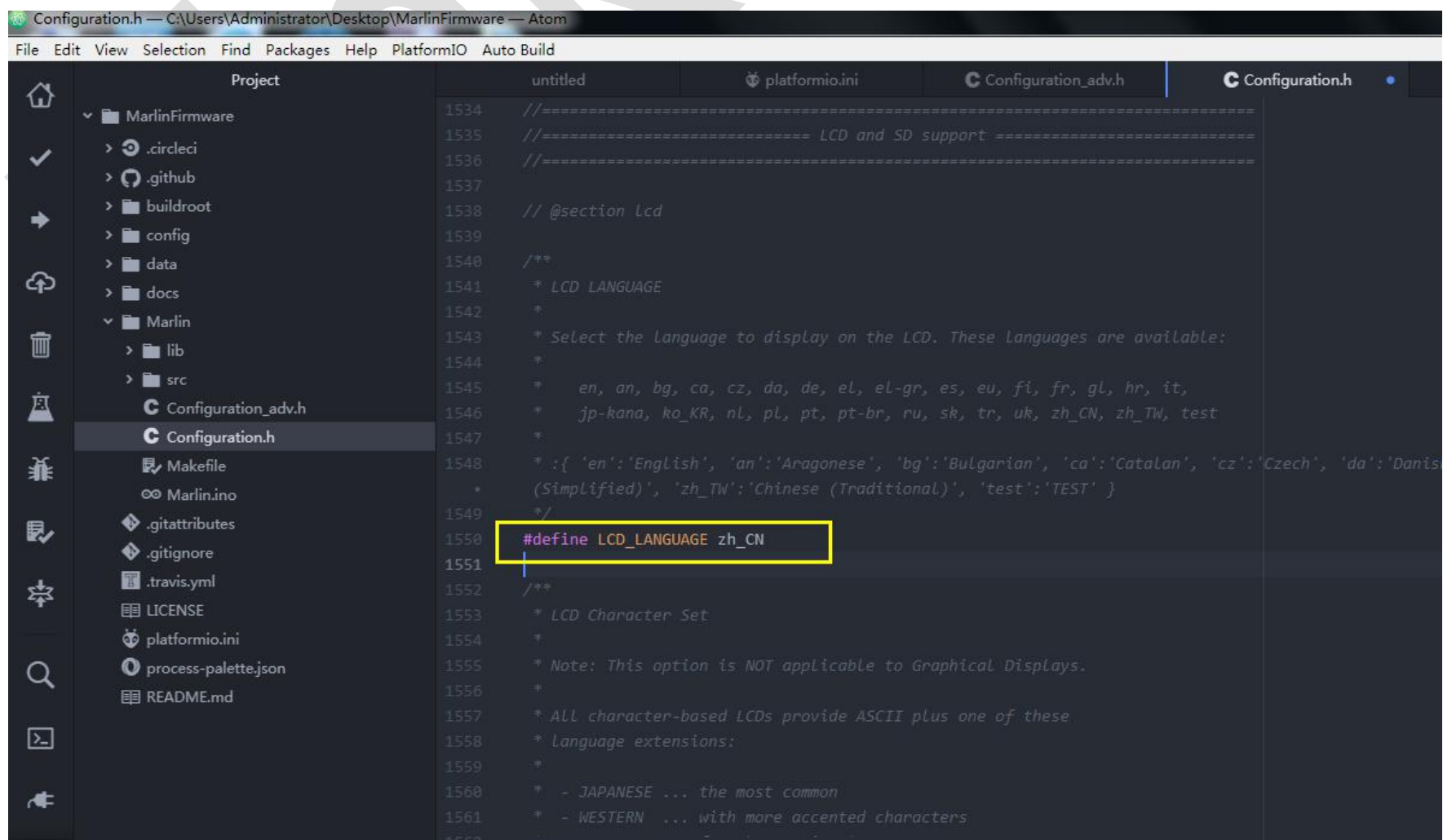
Travel limit: Set the maximum running distance of the xyz axis.



LCD screen setting

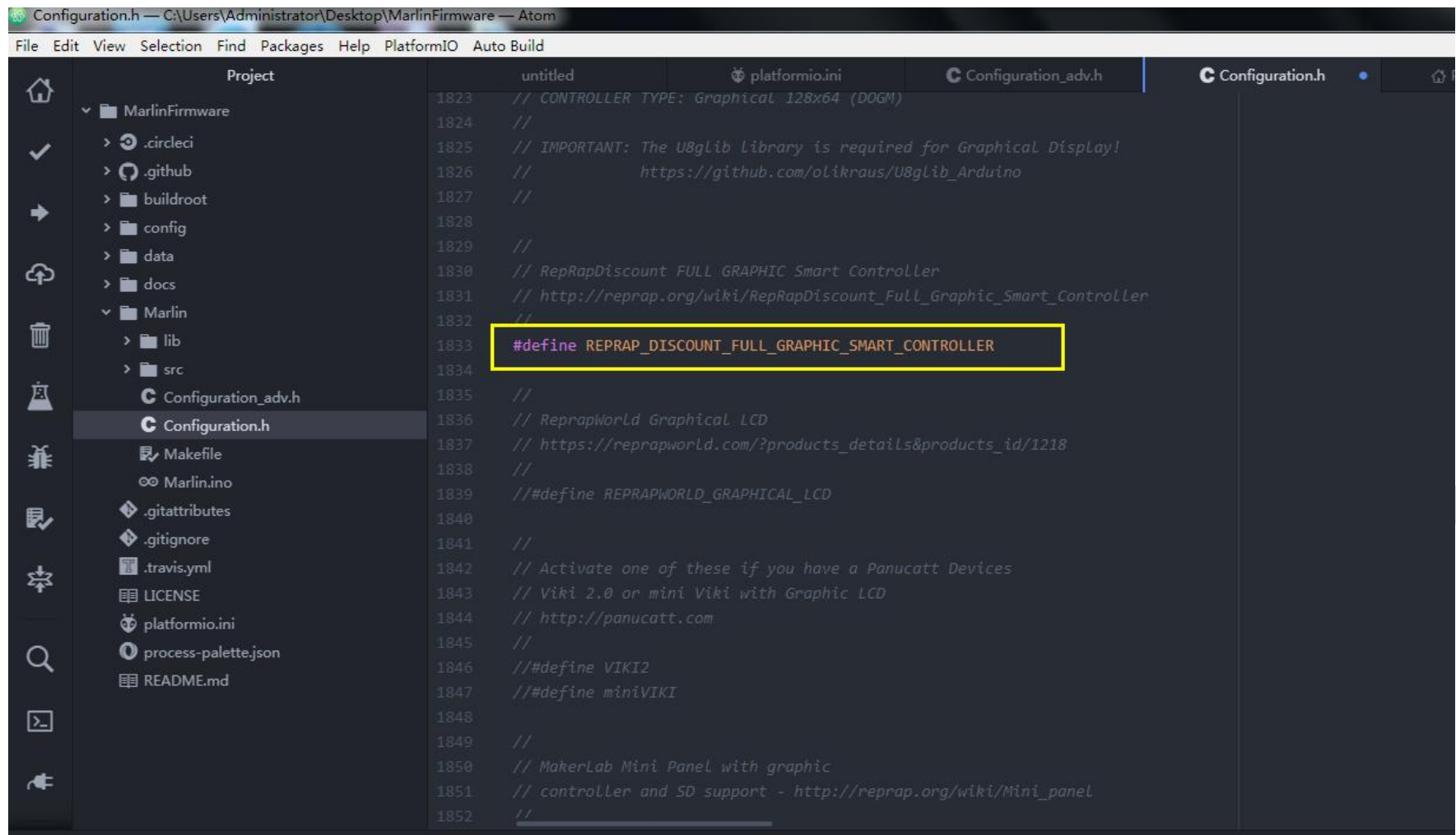
Language setting: zh_CN is Simplified Chinese, en is English.

Language settings can be made according to your needs.

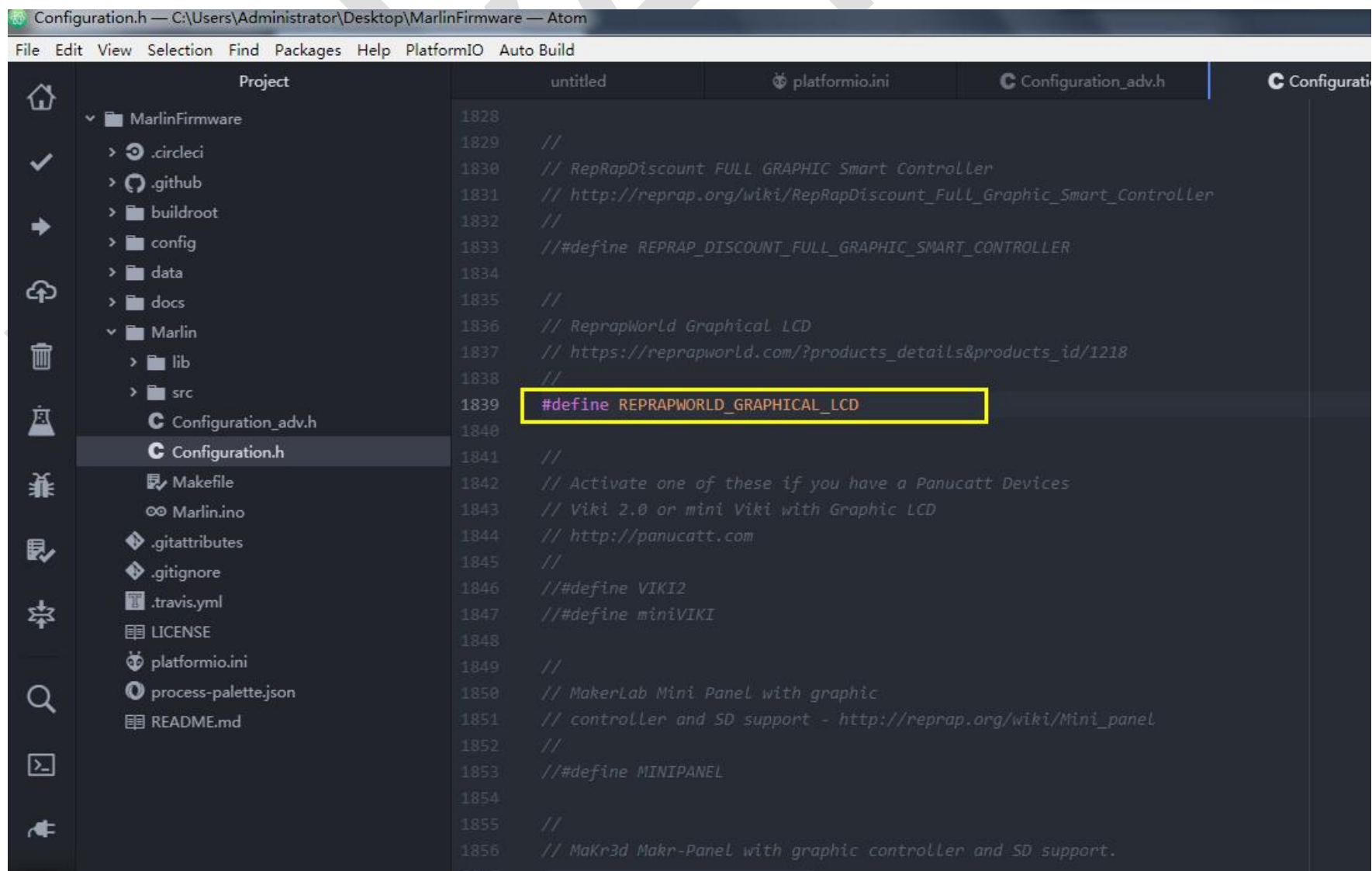


Screen type enable

Do not enable multiple screens at the same time. It is easy to cause errors LCD12864 enabled

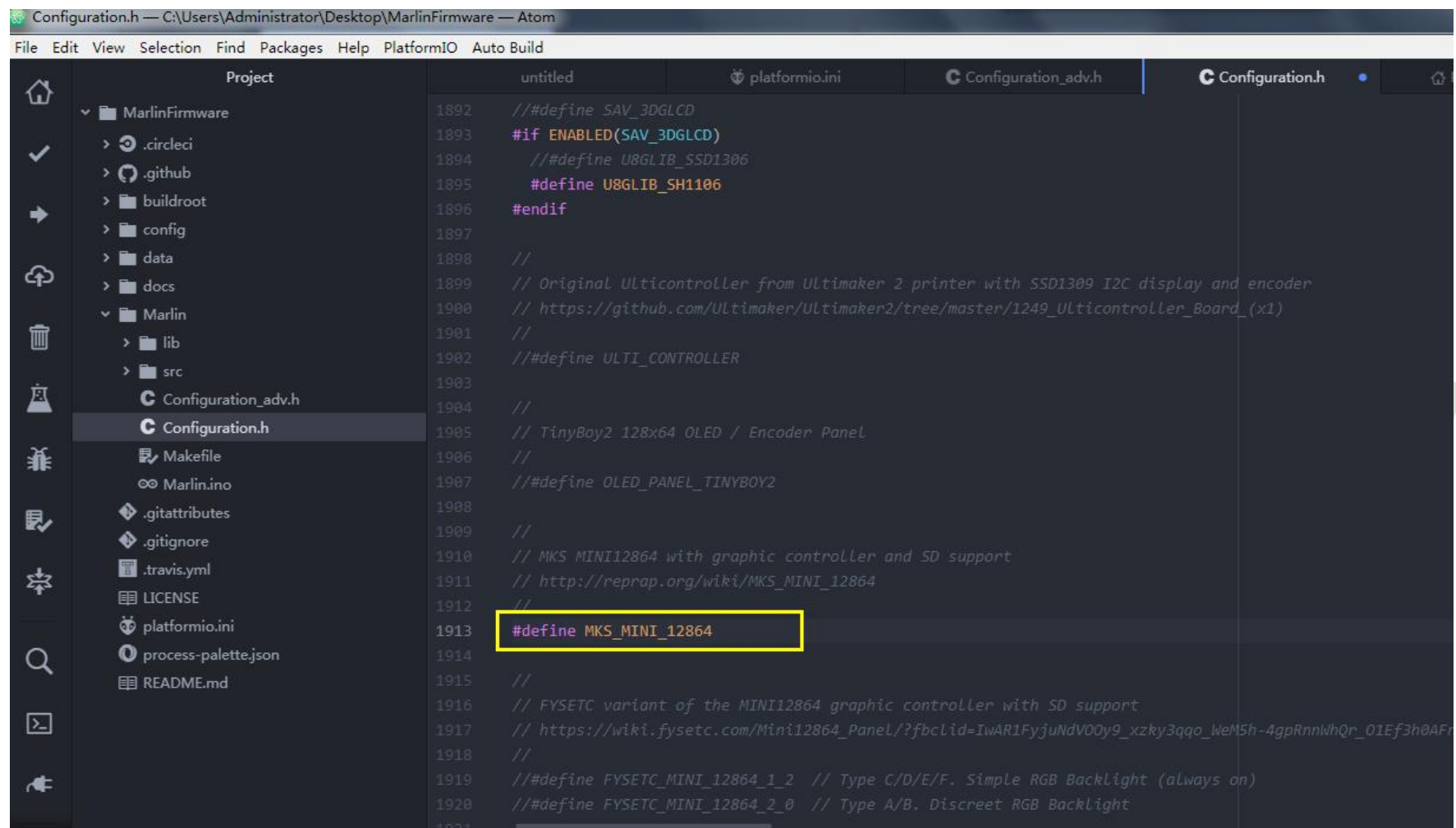


LCD2004 enable



Mini12864 and LCD12864A enable

If you use the 12864A, you need to modify the contrast of the screen in the firmware before uploading the file. Can be used again. (mini12864 does not need this operation)



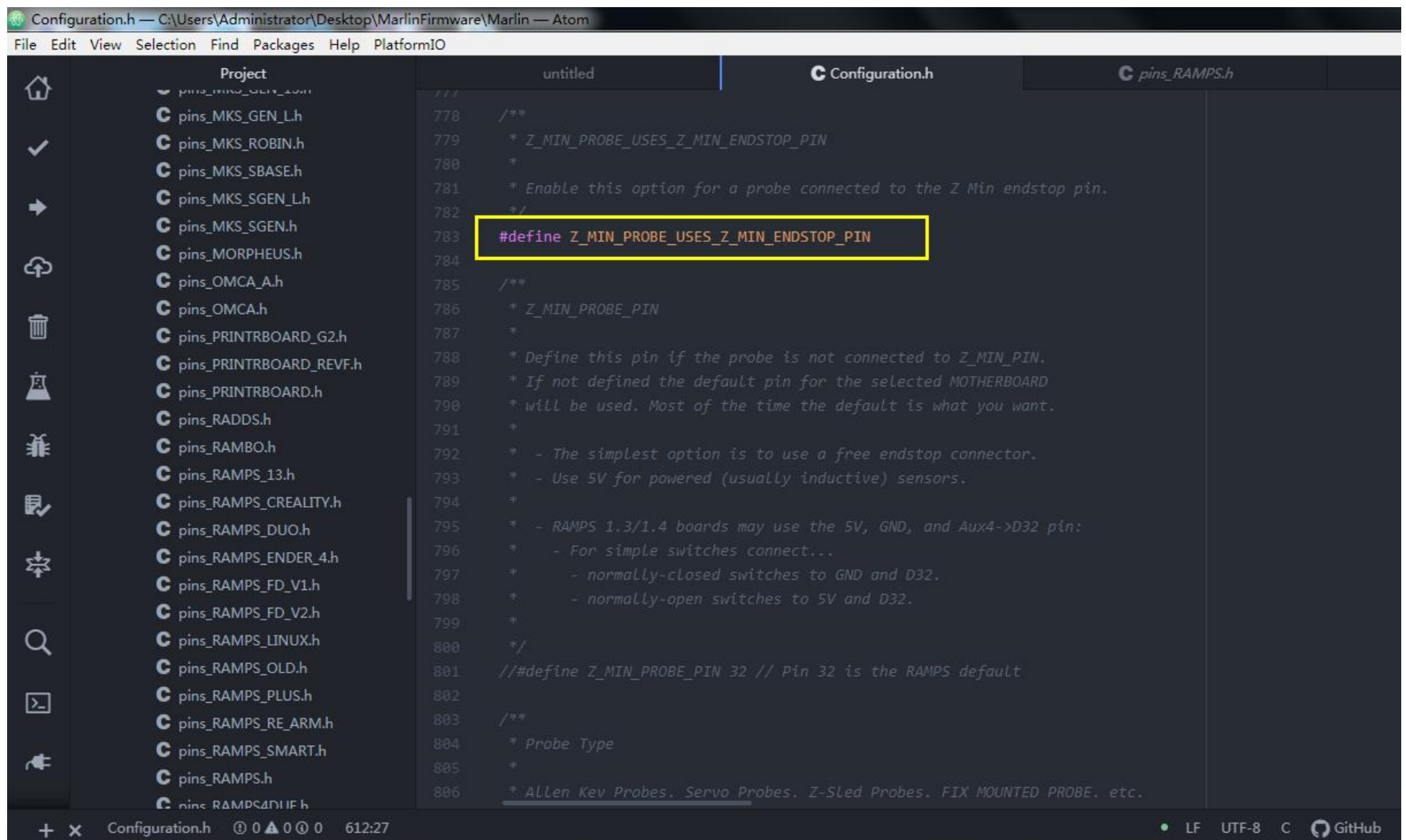
```
1892 // #define SAV_3DGLCD
1893 #if ENABLED(SAV_3DGLCD)
1894 // #define U8GLIB_SSD1306
1895 #define U8GLIB_SH1106
1896 #endif
1897
1898 //
1899 // Original Ulticontroller from Ultimaker 2 printer with SSD1309 I2C display and encoder
1900 // https://github.com/Ultimaker/Ultimaker2/tree/master/1249_Ulticontroller_Board_(x1)
1901 //
1902 // #define ULTI_CONTROLLER
1903
1904 //
1905 // TinyBoy2 128x64 OLED / Encoder Panel
1906 //
1907 // #define OLED_PANEL_TINYBOY2
1908 //
1909 //
1910 // MKS MINI12864 with graphic controller and SD support
1911 // http://reprap.org/wiki/MKS_MINI_12864
1912 //
1913 #define MKS_MINI_12864
1914
1915 //
1916 // FYSETC variant of the MINI12864 graphic controller with SD support
1917 // https://wiki.fysetc.com/Mini12864_Panel/?fbclid=IwAR1FyjuNdV00y9_xzky3qqo_wem5h-4gpRnnlwhQr_01Ef3h0AFr
1918 //
1919 // #define FYSETC_MINI_12864_1_2 // Type C/D/E/F. Simple RGB Backlight (always on)
1920 // #define FYSETC_MINI_12864_2_0 // Type A/B. Discreet RGB Backlight
1921
```

5.2.2 Automatic Leveling Setting

The configuration on the Marlin 2.0 firmware mainly refers to the "3D touch datasheet"

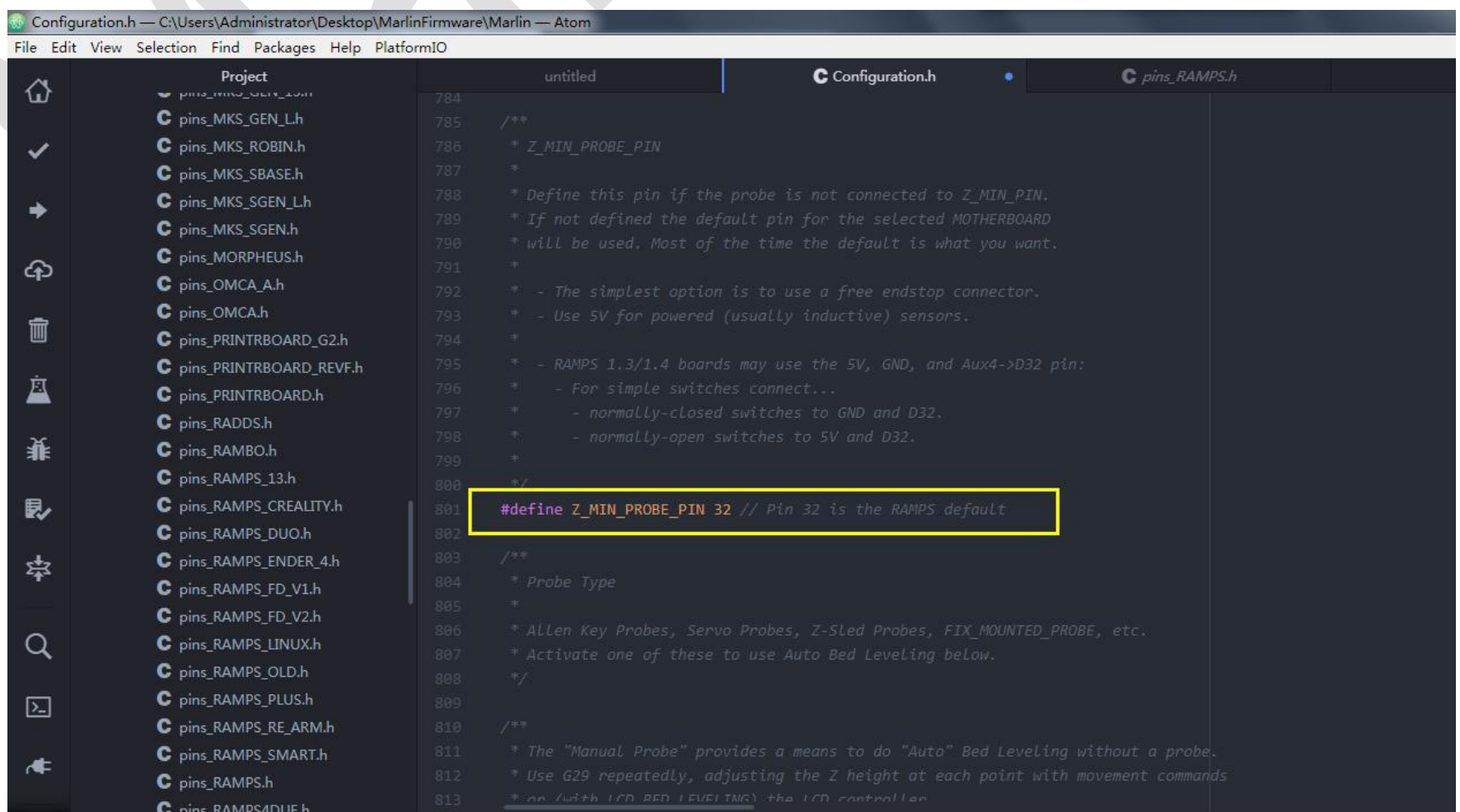
If using the Z-limit as the trigger limit for the leveling switch, enable the following item.

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

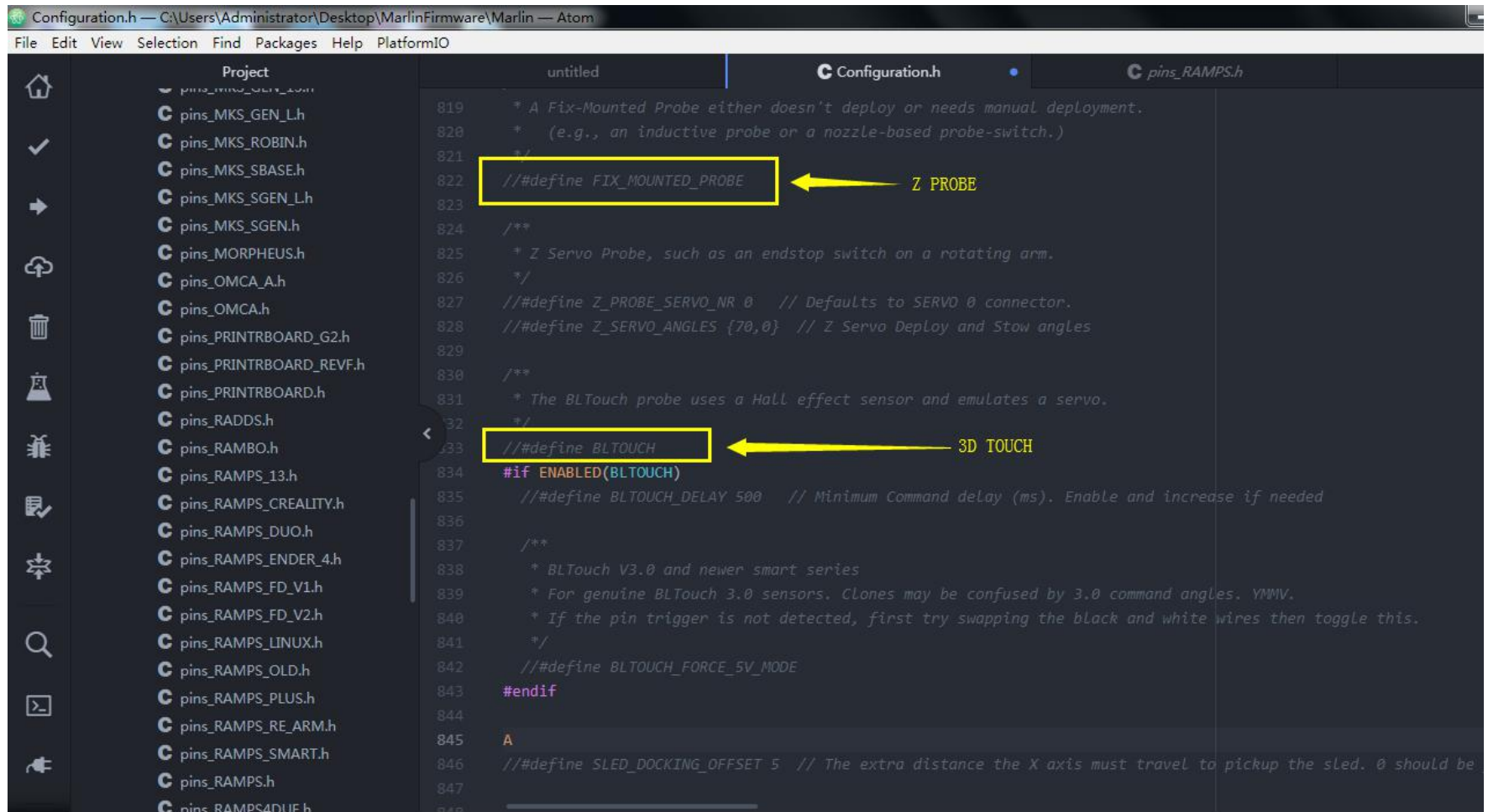


If you want to enable on other endstop's pin, you can enable the following item and modify the pin of the corresponding limit.

```
#define Z_MIN_PROBE_PIN 32
```

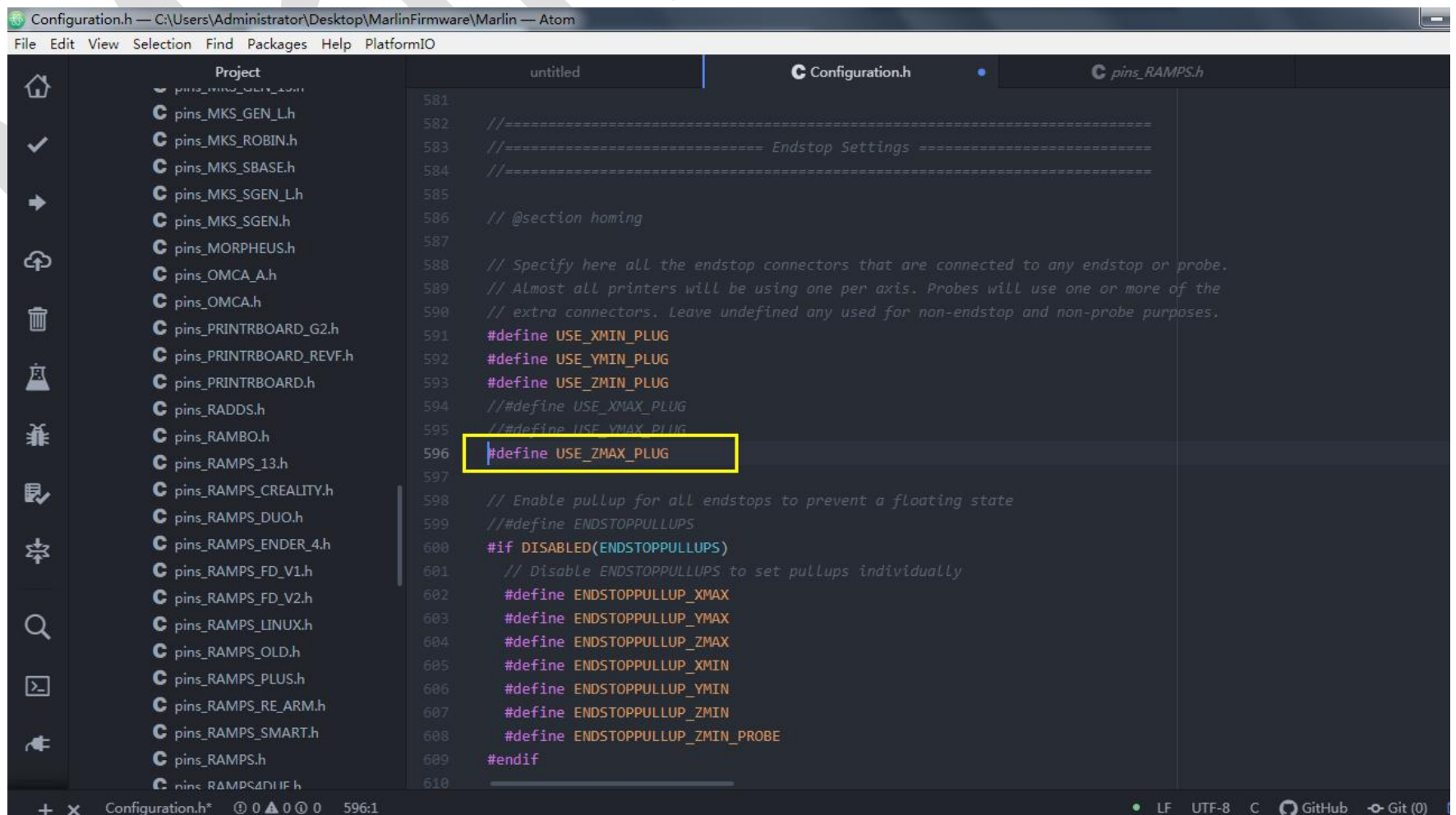


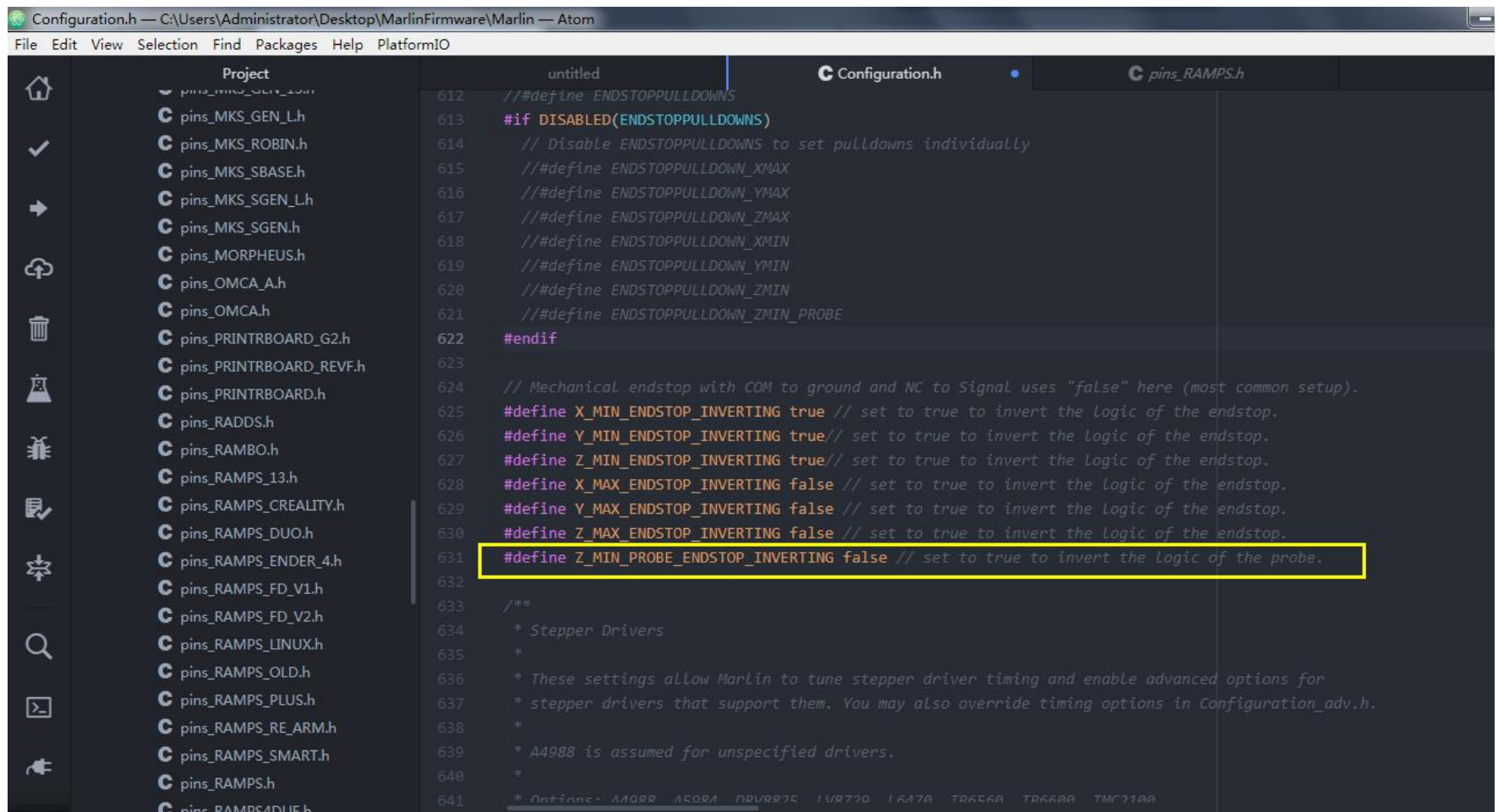
Look at the following figure, If you use Z-probe, enable the first item. If you use 3dtouch, enable the second one. Cannot enable multiple items at the same time, because it will cause errors. If you enable one item, you need to shield another one.



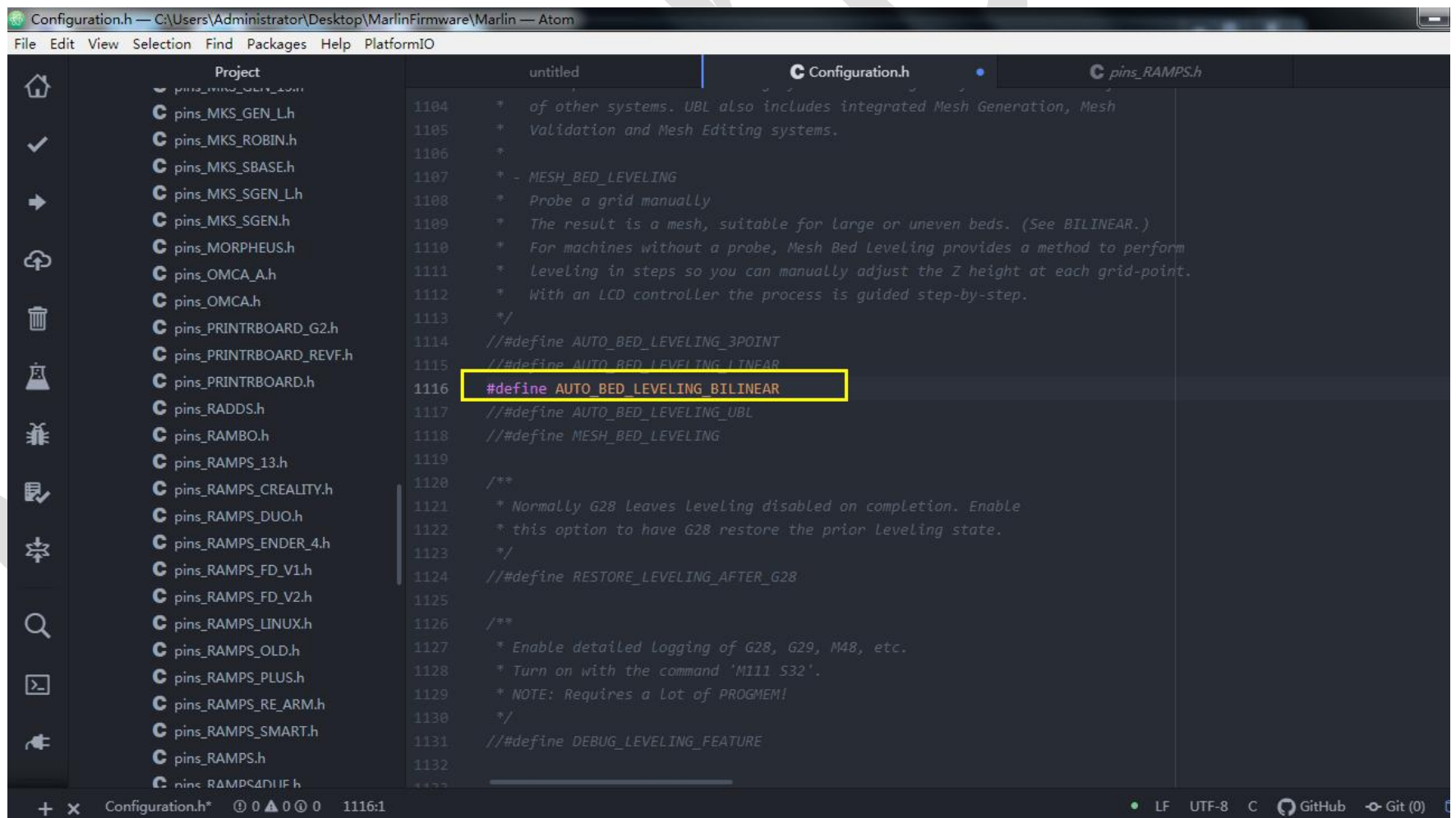
If the leveling switch is used with the relevant endstop, it must be enabled.

Generally, the default setting is false. Thus, the switch type (normally open or normally closed) should also be set correctly.

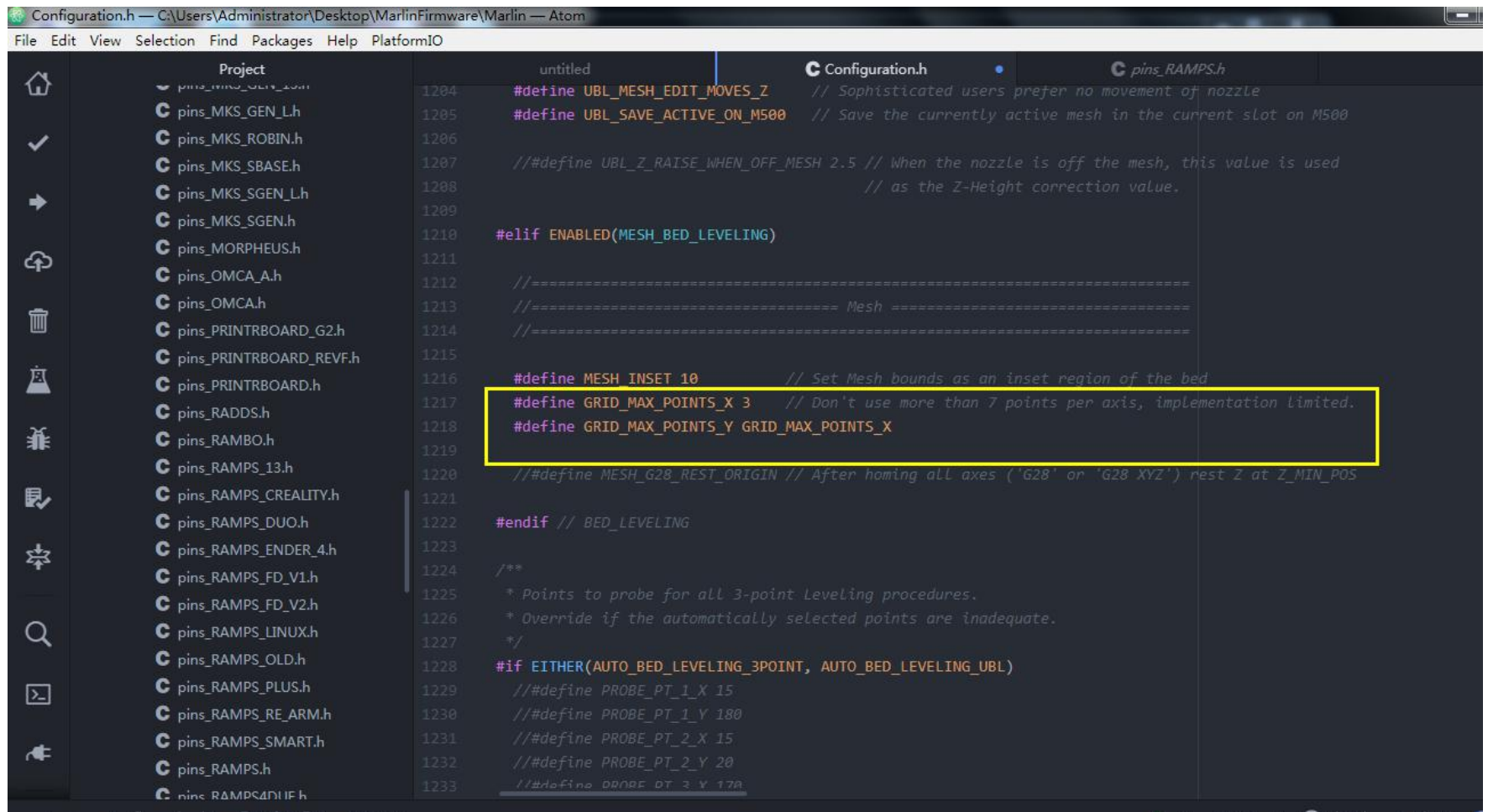




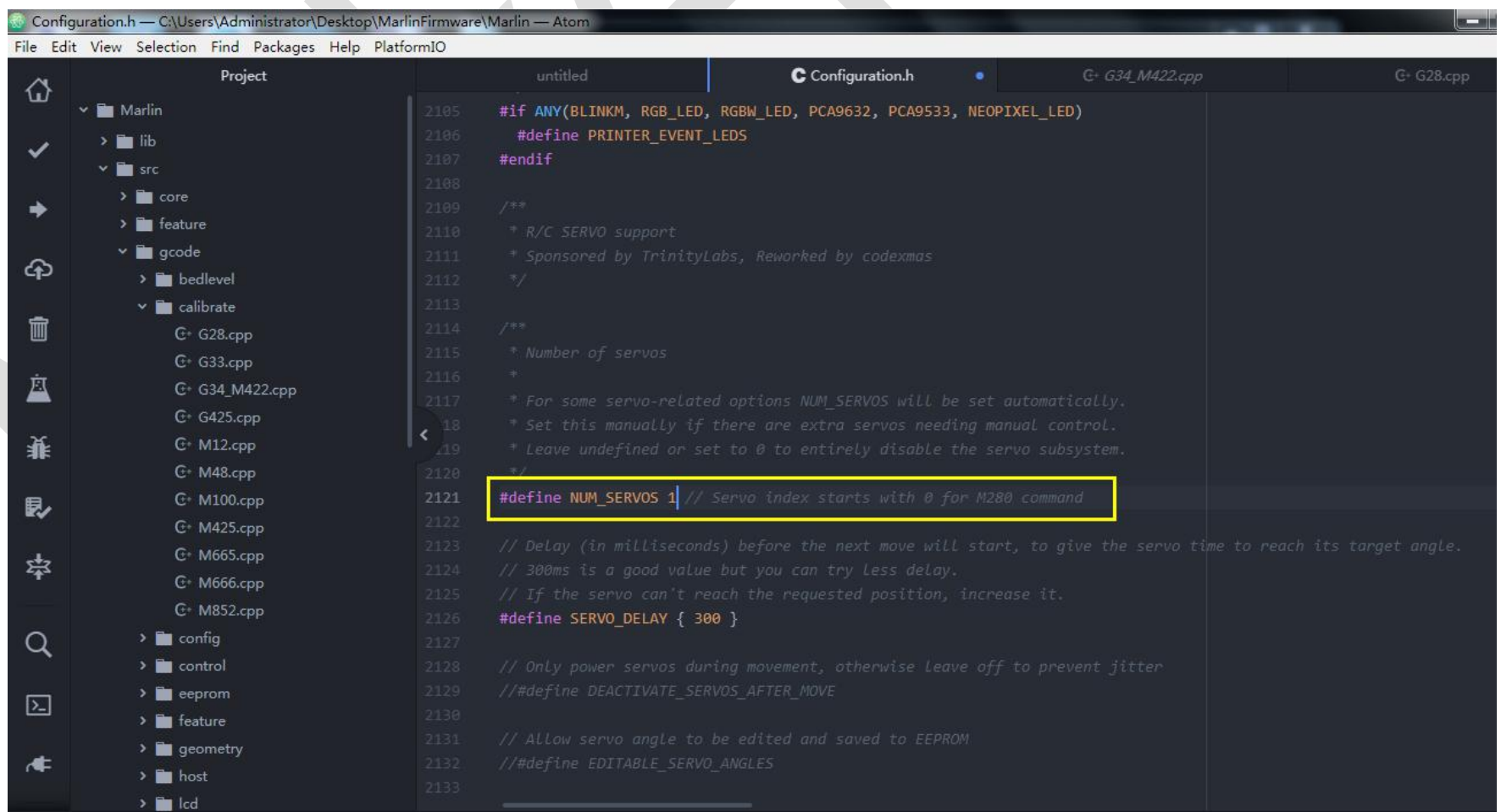
Select leveling mode, enable: #define AUTO_BED_LEVELING_BILINEAR



Leveling point setting



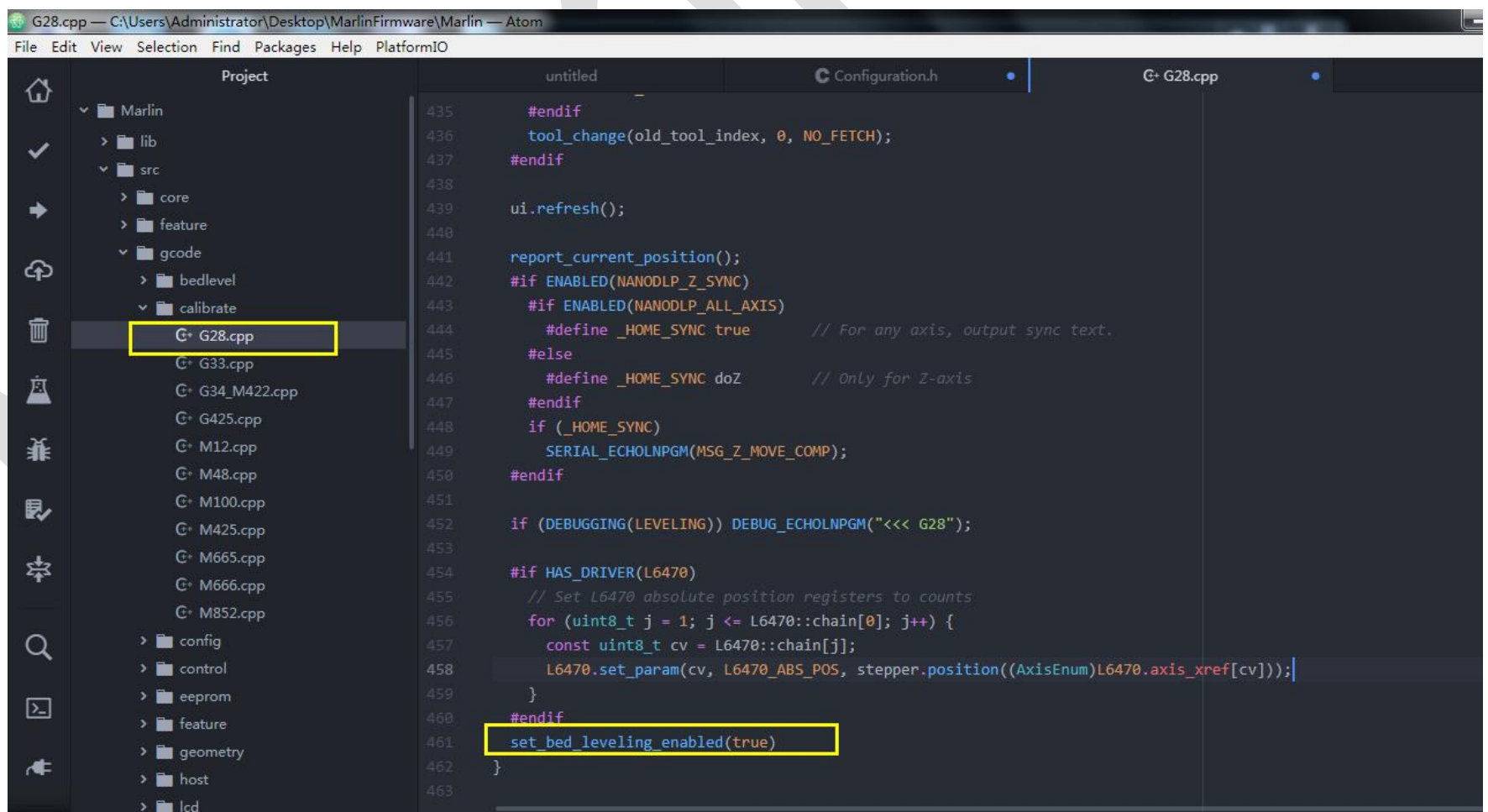
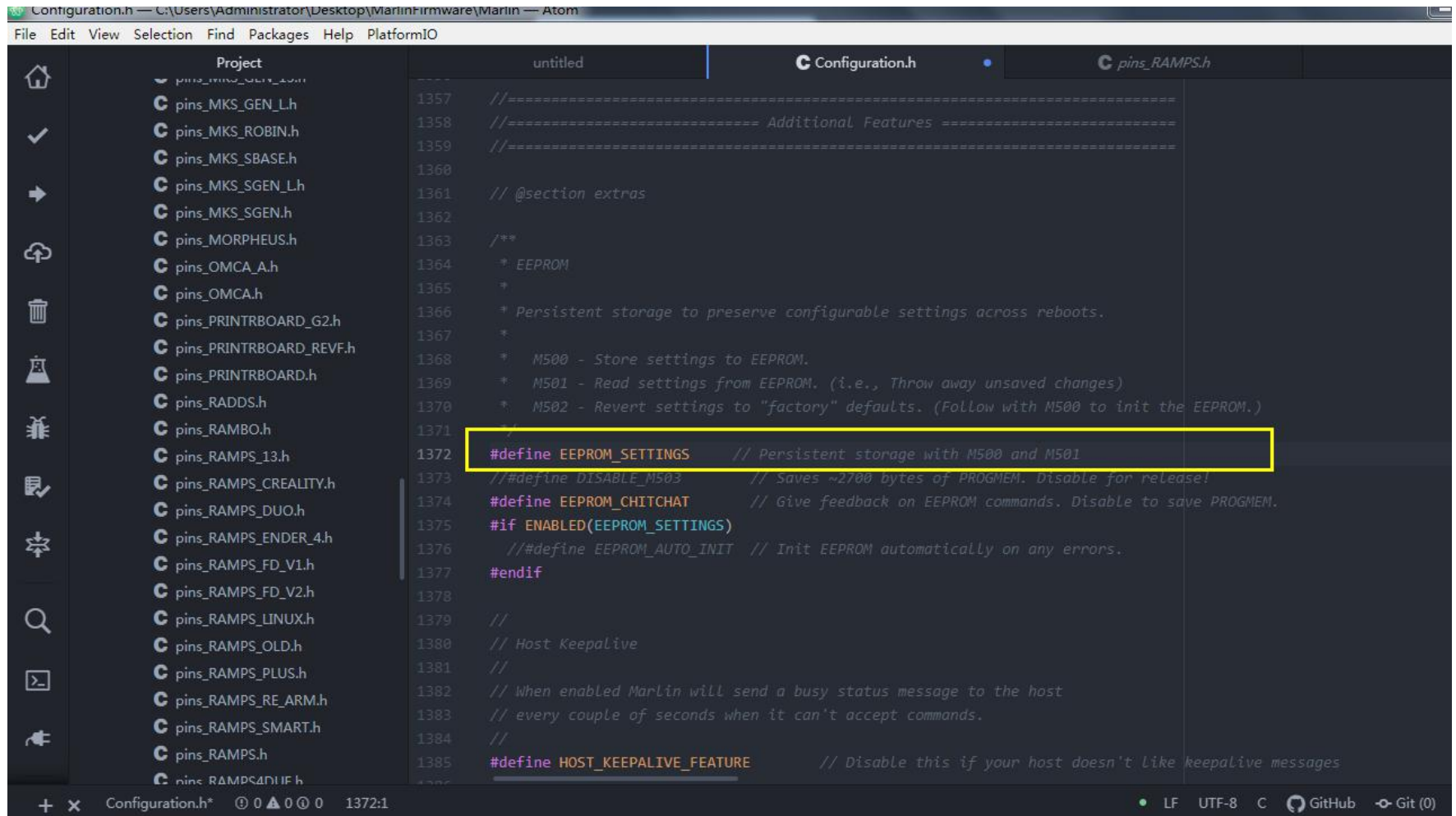
Change the number to “1”



Leveling data saving settings

Need to enable first: #define EEPROM_SETTINGS first

“Add set_bed_leveling_enabled(true)”to the “gocde- calibrate-G28.cpp”file, as the following figure shows.



Those are the auto-leveling setting.

5.2.3 Driver mode setting

Driver mode is composed by 3 types: normal mode, UART mode, and SPI mode

Note: the following driver setting is based on mks series of drivers, other manufacturers may have different driver setting steps.

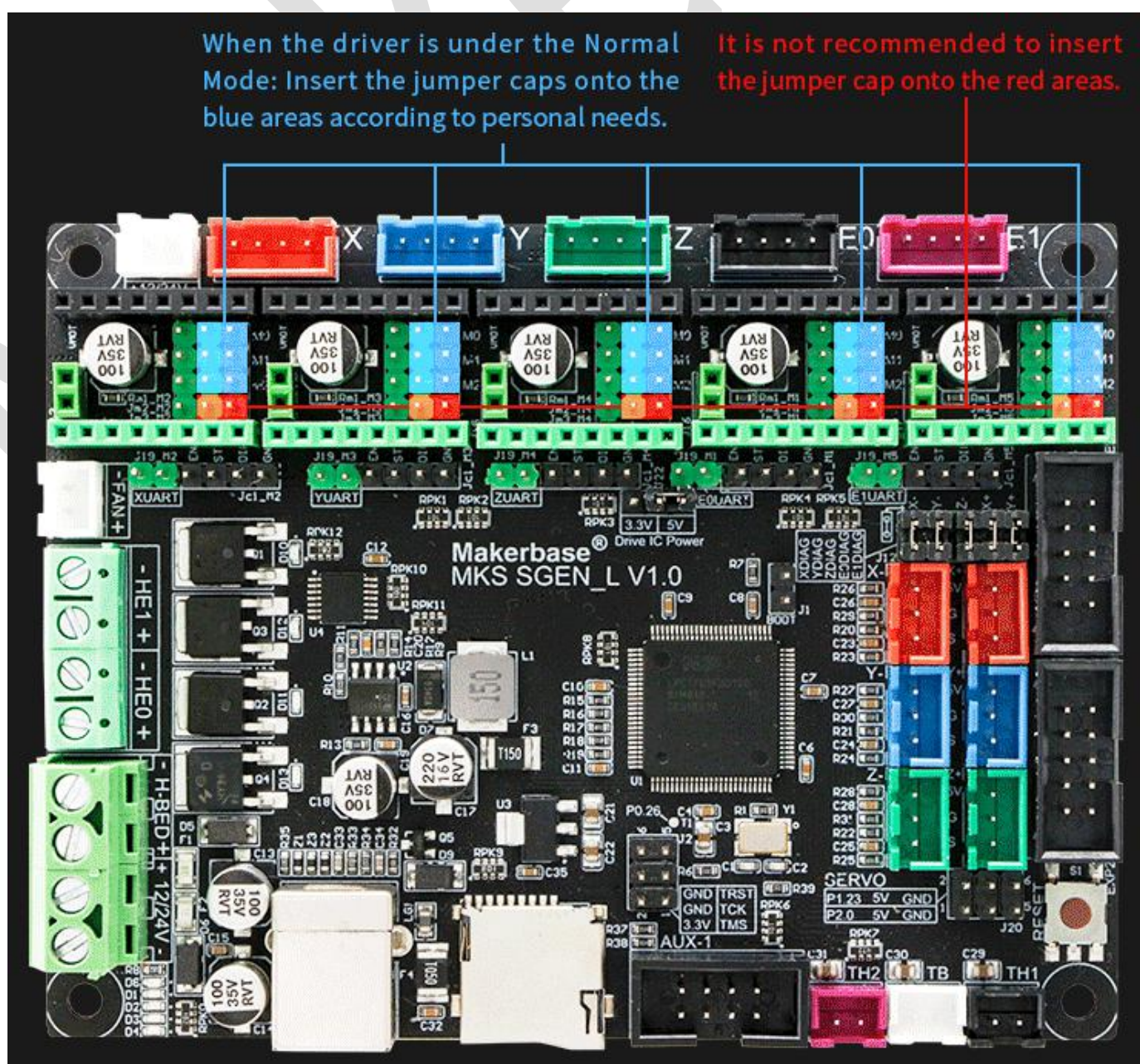
Normal mode (step/dir mode)

normal mode: Generally, it is available for all drivers.

Adjust the driving microstep by the jumper caps according to your own needs, the jumper caps can be inserted into the M0, M1, M2 of the two rows of black pin headers on the right side to adjust microsteps

Note: In the normal mode, the fourth pin doesn't need jumper caps, because wearing jumper caps may affect the normal use of some drivers. (eg TB67S109)

It is not recommended to insert jumper cap on the fourth pin, because it may affect the normal use of some drivers. (eg TB67S109)

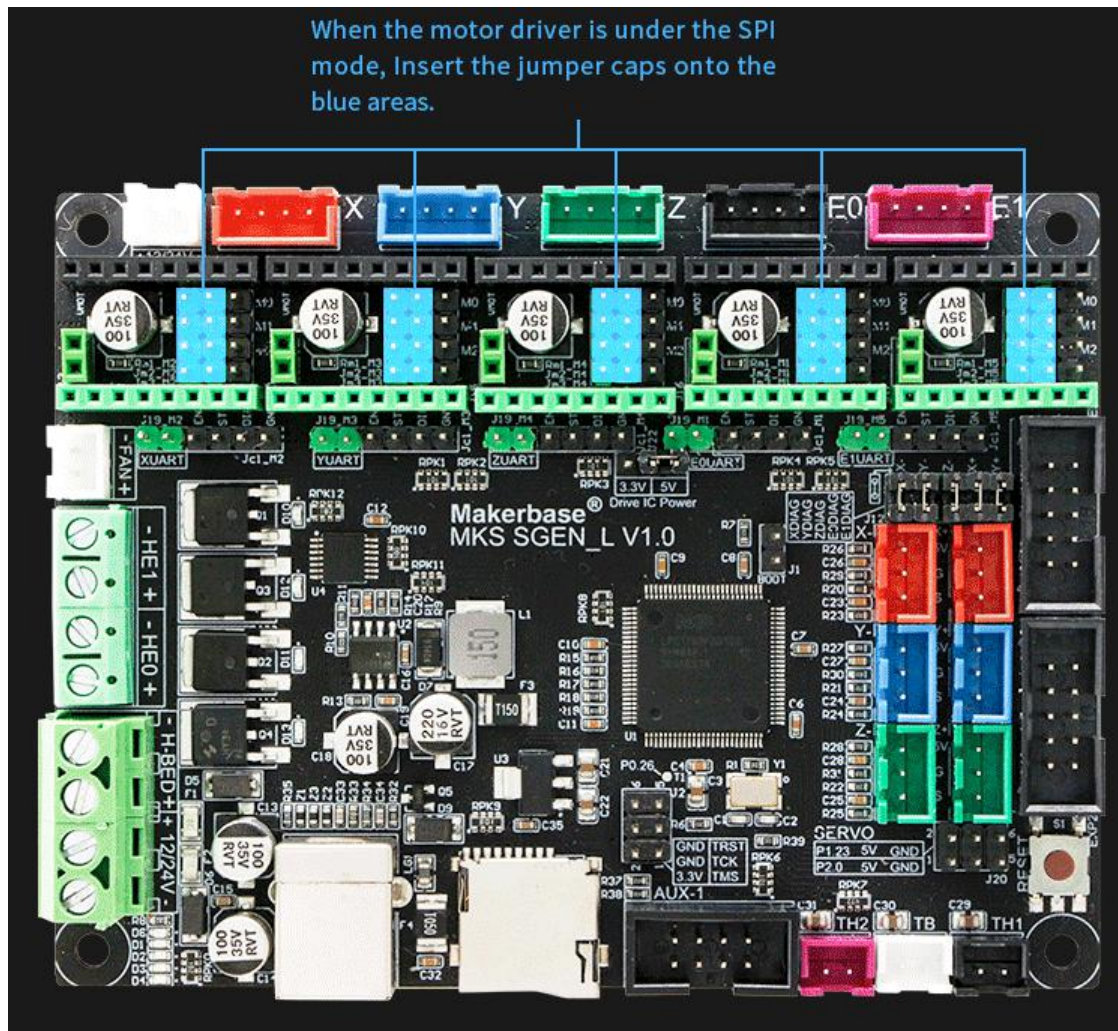


TMC2130 SPI mode setting

MKS TMC2130 SPI mode is composed of hardware and software setting.

Hardware setting: The motherboard jumper cap setting, only need to insert the jumper cap according to the requirements, no need jumper. as the picture shows

Just insert the four rows of jumper caps on the left side.



Software setting: configure SPI parameter in marlin firmware2.0, specific setting steps are as follow:

1. Driver type is set: TMC2130

```

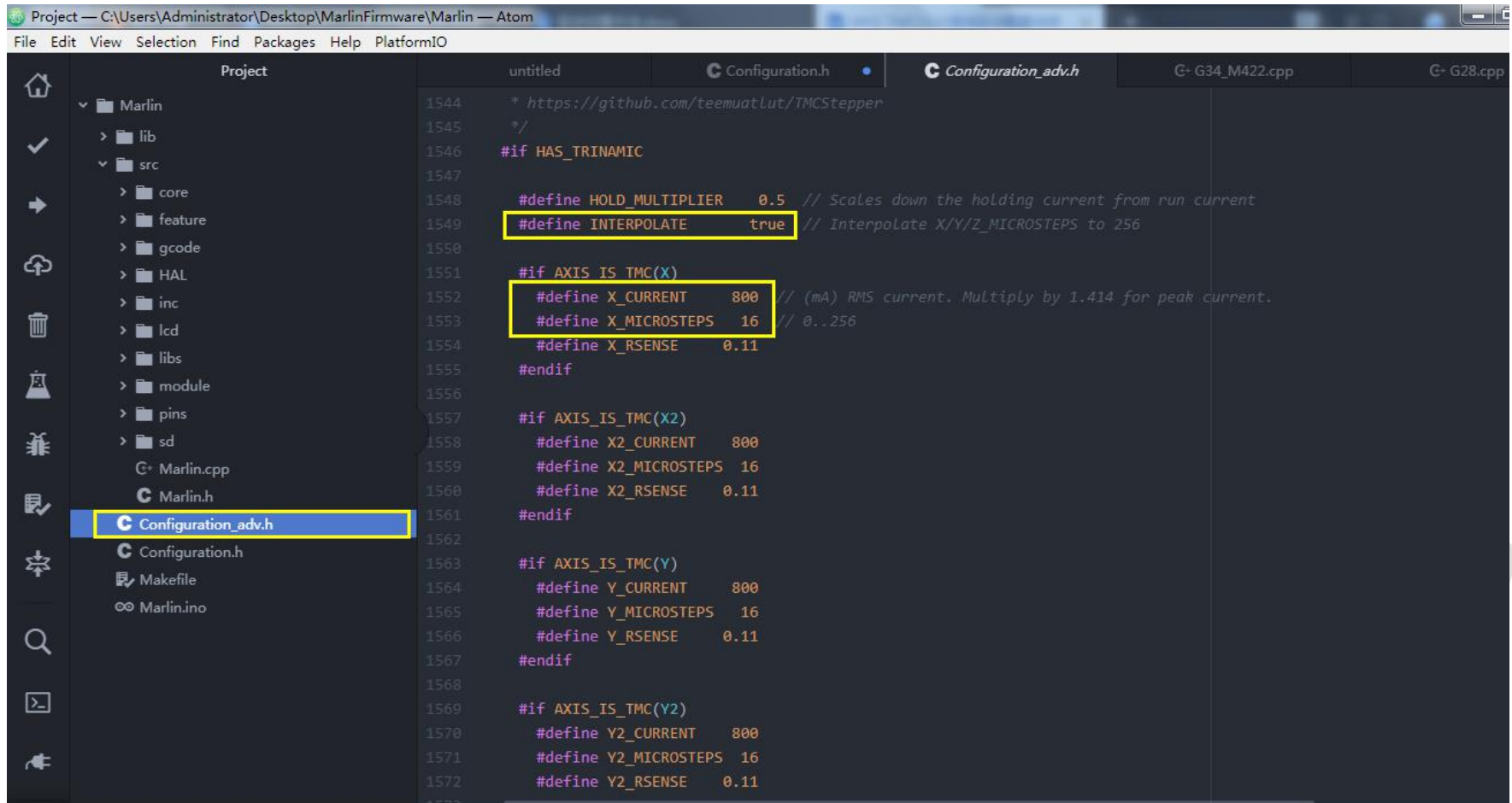
Project — C:\Users\Administrator\Desktop\MarlinFirmware\Marlin — Atom
File Edit View Selection Find Packages Help PlatformIO

Project: Marlin
lib
src
  core
  feature
  gcode
  HAL
  inc
  lcd
  libs
  module
  pins
  sd
  C+ Marlin.cpp
  C Marlin.h
  C Configuration_adv.h
  C Configuration.h
  Makefile
  Marlin.ino

untitled
Configuration.h
G+ G34_M422.cpp

640 *
641 * Options: A4988, A5984, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
642 *           TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
643 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
644 *           TMC2160, TMC2160_STANDALONE, TMC5130, TMC5130_STANDALONE,
645 *           TMC5160, TMC5160_STANDALONE
646 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130',
647 *
648 #define X_DRIVER_TYPE  TMC2130
649 #define Y_DRIVER_TYPE  TMC2130
650 #define Z_DRIVER_TYPE  TMC2130
651 // #define X2_DRIVER_TYPE A4988
652 // #define Y2_DRIVER_TYPE A4988
653 // #define Z2_DRIVER_TYPE TMC2130
654 // #define Z3_DRIVER_TYPE A4988
655 #define E0_DRIVER_TYPE TMC2130
656 // #define E1_DRIVER_TYPE A4988
657 // #define E2_DRIVER_TYPE A4988
658 // #define E3_DRIVER_TYPE A4988
659 // #define E4_DRIVER_TYPE A4988
660 // #define E5_DRIVER_TYPE A4988
661
662 // Enable this feature if all enabled endstop pins are interrupt-capable.
663 // This will remove the need to poll the interrupt pins, saving many CPU cycles.
664 // #define ENDSTOP_INTERRUPTS_FEATURE
665
666 /**
667 * Endstop Noise Threshold
    
```


2. Configure microstep and current in “Configuration_adv.h”



```

1544 * https://github.com/teemuatlut/TMCStepper
1545 */
1546 #if HAS_TRINAMIC
1547
1548 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
1549 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
1550
1551 #if AXIS_IS_TMC(X)
1552 #define X_CURRENT 800 // (mA) RMS current. Multiply by 1.414 for peak current.
1553 #define X_MICROSTEPS 16 // 0..256
1554 #define X_RSENSE 0.11
1555 #endif
1556
1557 #if AXIS_IS_TMC(X2)
1558 #define X2_CURRENT 800
1559 #define X2_MICROSTEPS 16
1560 #define X2_RSENSE 0.11
1561 #endif
1562
1563 #if AXIS_IS_TMC(Y)
1564 #define Y_CURRENT 800
1565 #define Y_MICROSTEPS 16
1566 #define Y_RSENSE 0.11
1567 #endif
1568
1569 #if AXIS_IS_TMC(Y2)
1570 #define Y2_CURRENT 800
1571 #define Y2_MICROSTEPS 16
1572 #define Y2_RSENSE 0.11
1573

```

Microstep setting: X_MICROSTEPS, default setting is 16 microsteps, because after you enable #define

INTERPOLATE true, the chip internally will expand the 16 microsteps into 256 microsteps

Current setting: #define X_CURRENT, “800” is current value, TMC2130 driving current is 500-600mA. Current can’t be over 600mA because the driver will warm up seriously if the heat dissipation condition is not good. If the heat dissipation condition is good (heat sink and heat dissipation fan) it can be exceeded 600mA, but the maximum value can’t be exceeded 1000mA. If temperature is too high, it can affect the operation of the chip and may result in losing steps.

The actual driving current value is 1.414 times as the setting one, so care should be taken not to exceed the maximum current of the driver or motor.

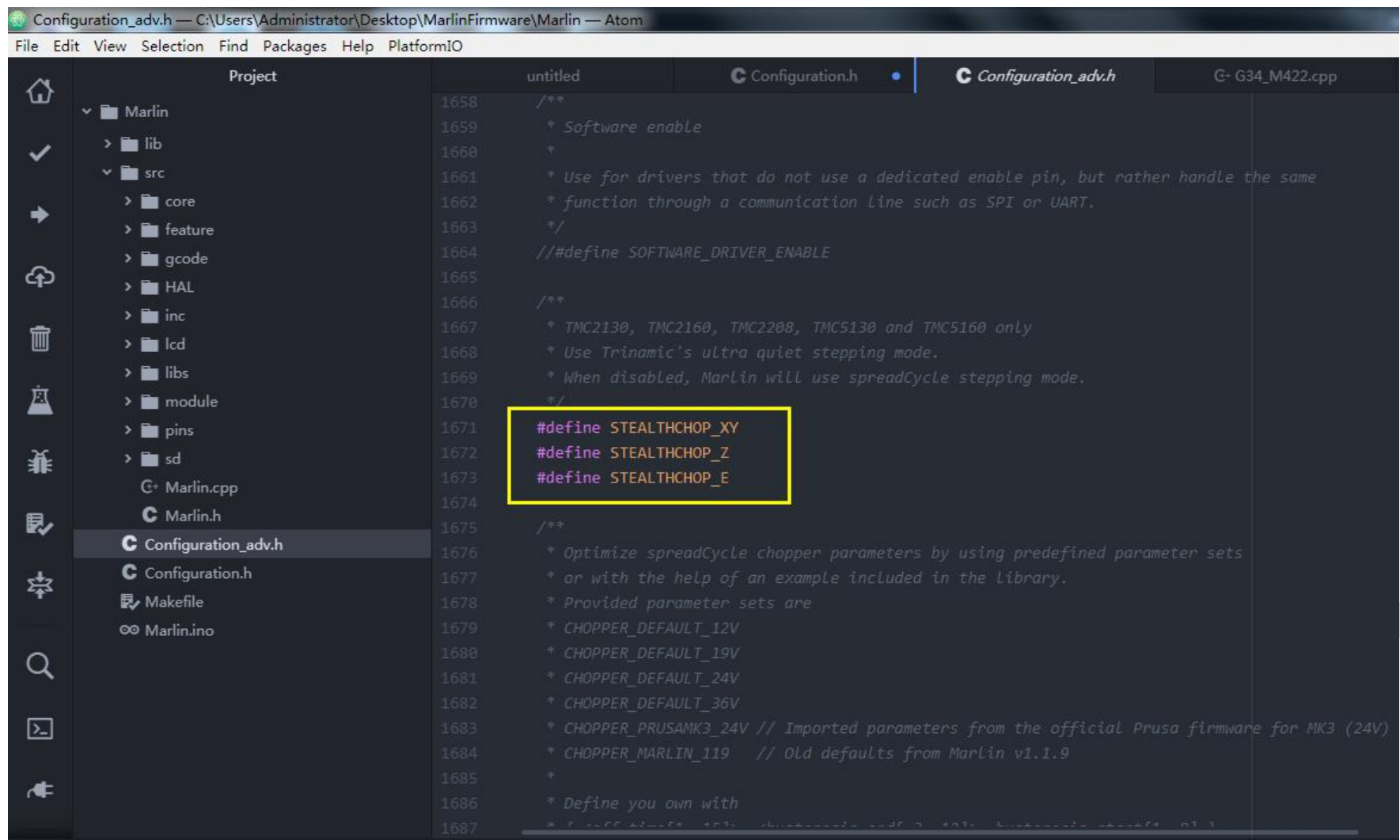
For example: the picture shows that the current is 800mA, but the actual running current $I=1.414*800\text{mA}$.

3. Operating mode selection

The SPI mode adopts software IO to simulate SPI mode, which does not need to define the hardware SPI mode in the firmware.

Configure parameter in the Configuration_adv.h file

Enable “STEALTHCHOP” : silent mode. If you delete “STEALTHCHOP” (double slash before the item //): the spreadCycle mode.

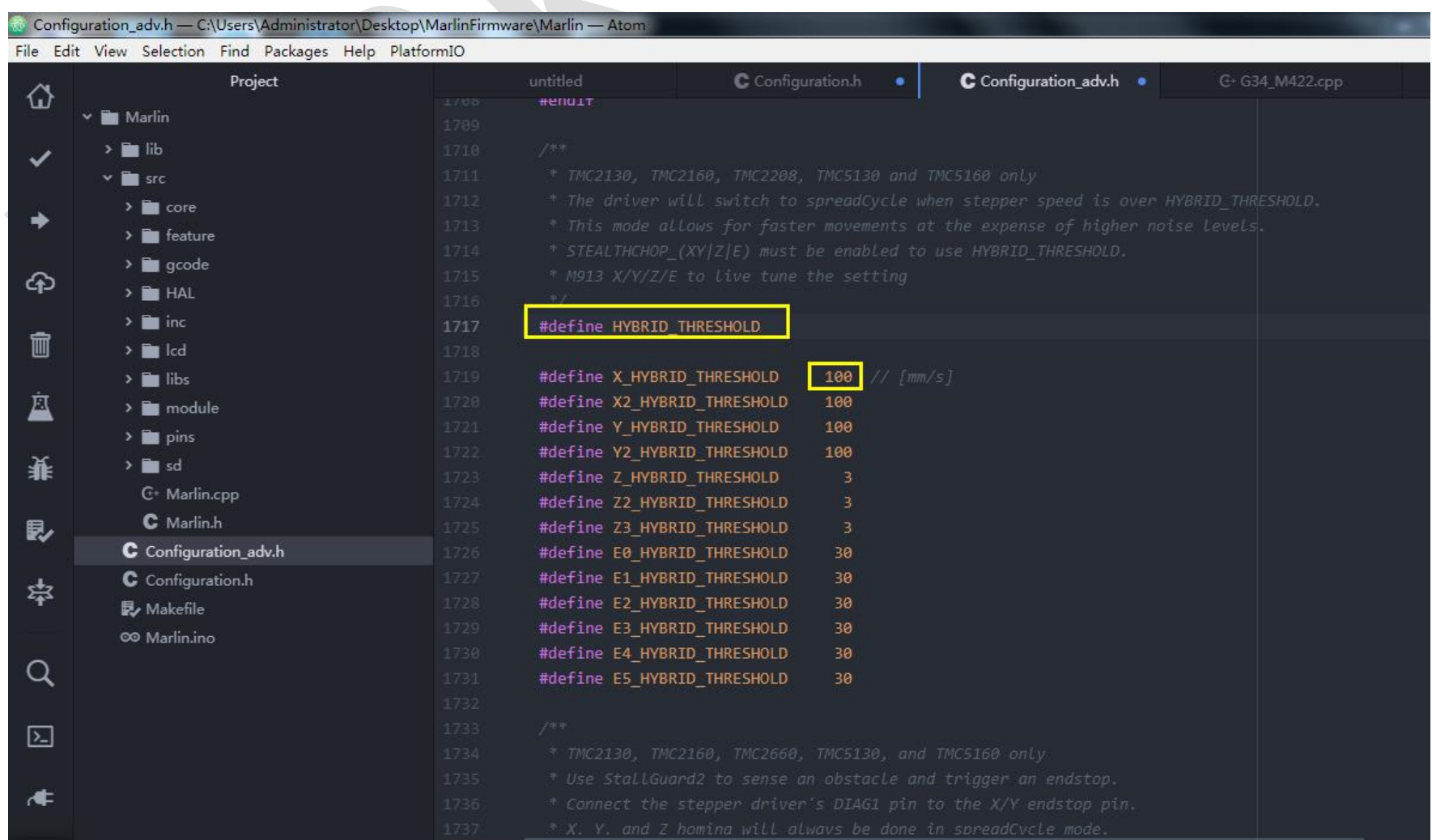


If you want to configure the Hybrid mode, you must first enable the STEALTHCHOP (silent mode)

Enable #define HYBRID_THRESHOLD

The corresponding value after “HYBRID_THRESHOLD” is the value of the mode switching. If the motor runs 100mm/s faster than the setting value, it will switch to the spreadcycle mode, if below 100mm/s, it will be the stealthchop mode.

Mixed mode can be configured according to your own needs



4.CS pin setting

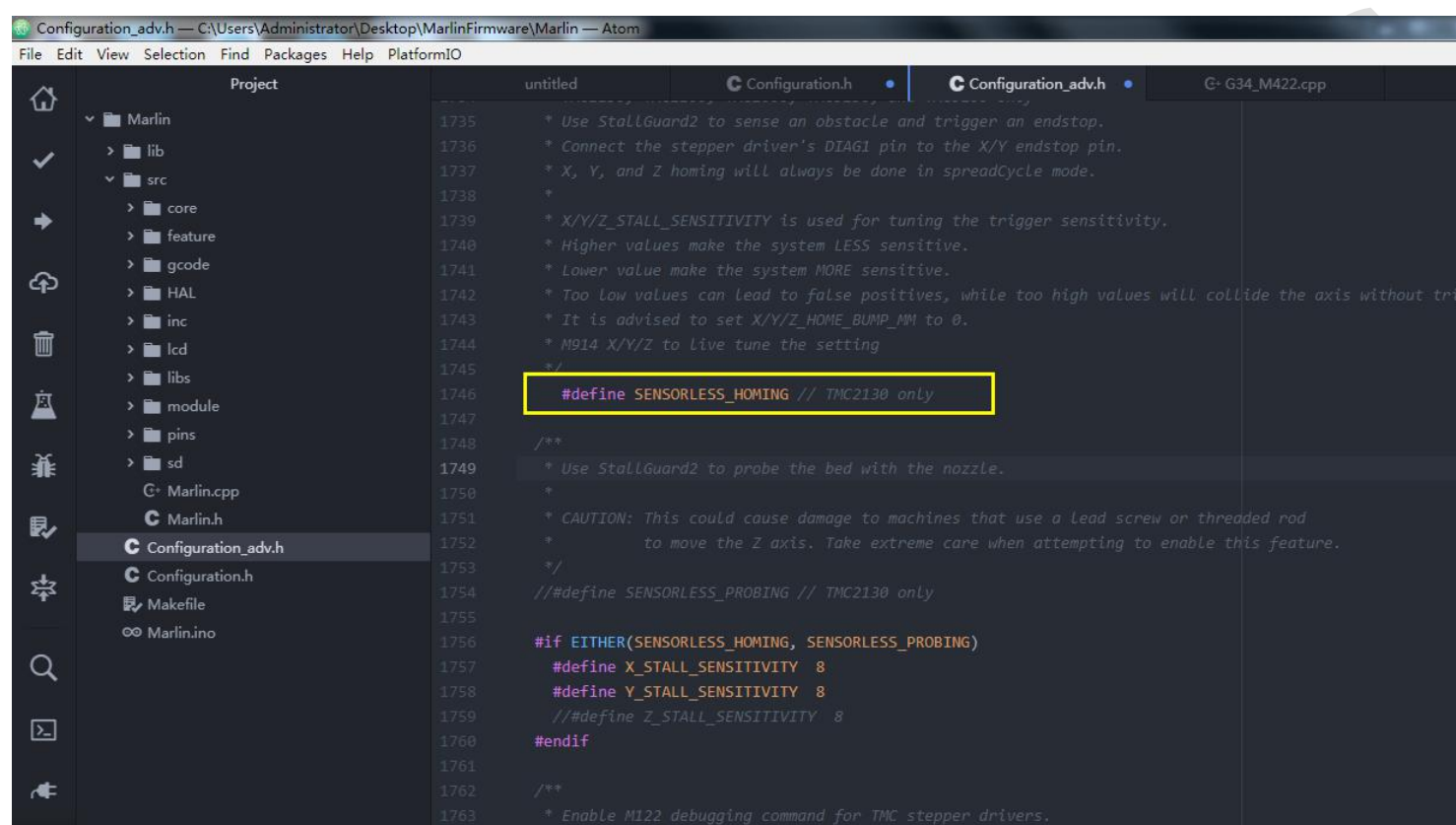
Because the wires are integrated internally to SGEN_L, and the pin file has been set, which doesn't need to be modified.

Homing without limit setting

5.Homing without limit configuration (different to configure, Optionally)

If you use the ordinary mechanical limit, you can not configure this item.

Enable #define SENSORLESS_HOMING in the "Configuration_adv.h"



```

1735  * Use StallGuard2 to sense an obstacle and trigger an endstop.
1736  * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
1737  * X, Y, and Z homing will always be done in spreadCycle mode.
1738  *
1739  * X/Y/Z_STALL_SENSITIVITY is used for tuning the trigger sensitivity.
1740  * Higher values make the system LESS sensitive.
1741  * Lower values make the system MORE sensitive.
1742  * Too Low values can lead to false positives, while too high values will collide the axis without triggering.
1743  * It is advised to set X/Y/Z_HOME_BUMP_MM to 0.
1744  * M914 X/Y/Z to live tune the setting
1745  */
1746  #define SENSORLESS_HOMING // TMC2130 only
1747  /**
1748  * Use StallGuard2 to probe the bed with the nozzle.
1749  *
1750  * CAUTION: This could cause damage to machines that use a lead screw or threaded rod
1751  *           to move the Z axis. Take extreme care when attempting to enable this feature.
1752  */
1753  // #define SENSORLESS_PROBING // TMC2130 only
1754
1755  #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
1756    #define X_STALL_SENSITIVITY  8
1757    #define Y_STALL_SENSITIVITY  8
1758    // #define Z_STALL_SENSITIVITY  8
1759  #endif
1760
1761  /**
1762  * Enable M122 debugging command for TMC stepper drivers.
1763  */

```

Note: The wire of the driver DIAG1 pin needs to be connected to the X / Y limit pin. (Z axis is temporarily not supported)

using the TMC2130 SGL version doesn't need jumper.

X, Y and Z homing will only be done in spreadCycle mode.

X / Y / Z_HOMING_SENSITIVITY is used to adjust the trigger sensitivity. higher the value is, lower the system sensitivity will be. Lower values make the system more sensitive. If the value that is too low, it can cause error, while if the value that is too high it may hit the axis without triggering.

It is recommended to set X / Y / Z_HOME_BUMP_MM to 0.


```

506 #define DEFAULT_DUPLICATION_X_OFFSET 100
507
508 #endif // DUAL_X_CARRIAGE
509
510 // Activate a solenoid on the active extruder with M380. Disable all with M381.
511 // Define SOL0_PIN, SOL1_PIN, etc., for each extruder that has a solenoid.
512 //#define EXT_SOLENOID
513
514 // @section homing
515
516 // Homing hits each endstop, retracts by these distances, then does a slower bump.
517 #define X_HOME_BUMP_MM 0
518 #define Y_HOME_BUMP_MM 0
519 #define Z_HOME_BUMP_MM 2
520 #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
521 // #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
522 // #define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
523
524 // When G28 is called, this option will make Y home before X
525 // #define HOME_Y_BEFORE_X
526
527 // Enable this if X or Y can't home without homing the other axis first.
528 // #define CODEPENDENT_XY_HOMING
529
530 /**
531  * Z Steppers Auto-Alignment
532  * Add the G34 command to align multiple Z steppers using a bed probe.
533  */
534 // #define Z_STEPPER_AUTO_ALIGN

```

Note: Since the logic signal of the blocking detection processed by the TMC2130 is true, when the blocking detection is used as the limit, the limit level can only be set to true, otherwise the compilation will prompt an error; the Marlin firmware Z-axis cannot temporarily use the blocking detection as the For the limit position, the limit switch signal is also required when blocking detection is used as the limit. It is not recommended to use the block detection as the limit.

```

606 #define ENDSTOPPULLUP_YMIN
607 #define ENDSTOPPULLUP_ZMIN
608 #define ENDSTOPPULLUP_ZMIN_PROBE
609 #endif
610
611 // Enable pulldown for all endstops to prevent a floating state
612 // #define ENDSTOPPULLDOWNS
613 #if DISABLED(ENDSTOPPULLDOWNS)
614 // Disable ENDSTOPPULLDOWNS to set pulldowns individually
615 #define ENDSTOPPULLDOWN_XMAX
616 #define ENDSTOPPULLDOWN_YMAX
617 #define ENDSTOPPULLDOWN_ZMAX
618 // #define ENDSTOPPULLDOWN_XMIN
619 // #define ENDSTOPPULLDOWN_YMIN
620 // #define ENDSTOPPULLDOWN_ZMIN
621 // #define ENDSTOPPULLDOWN_ZMIN_PROBE
622 #endif
623
624 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup)
625 #define X_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
626 #define Y_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
627 #define Z_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
628 #define X_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
629 #define Y_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
630 #define Z_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
631 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // set to true to invert the logic of the probe.
632
633 /**

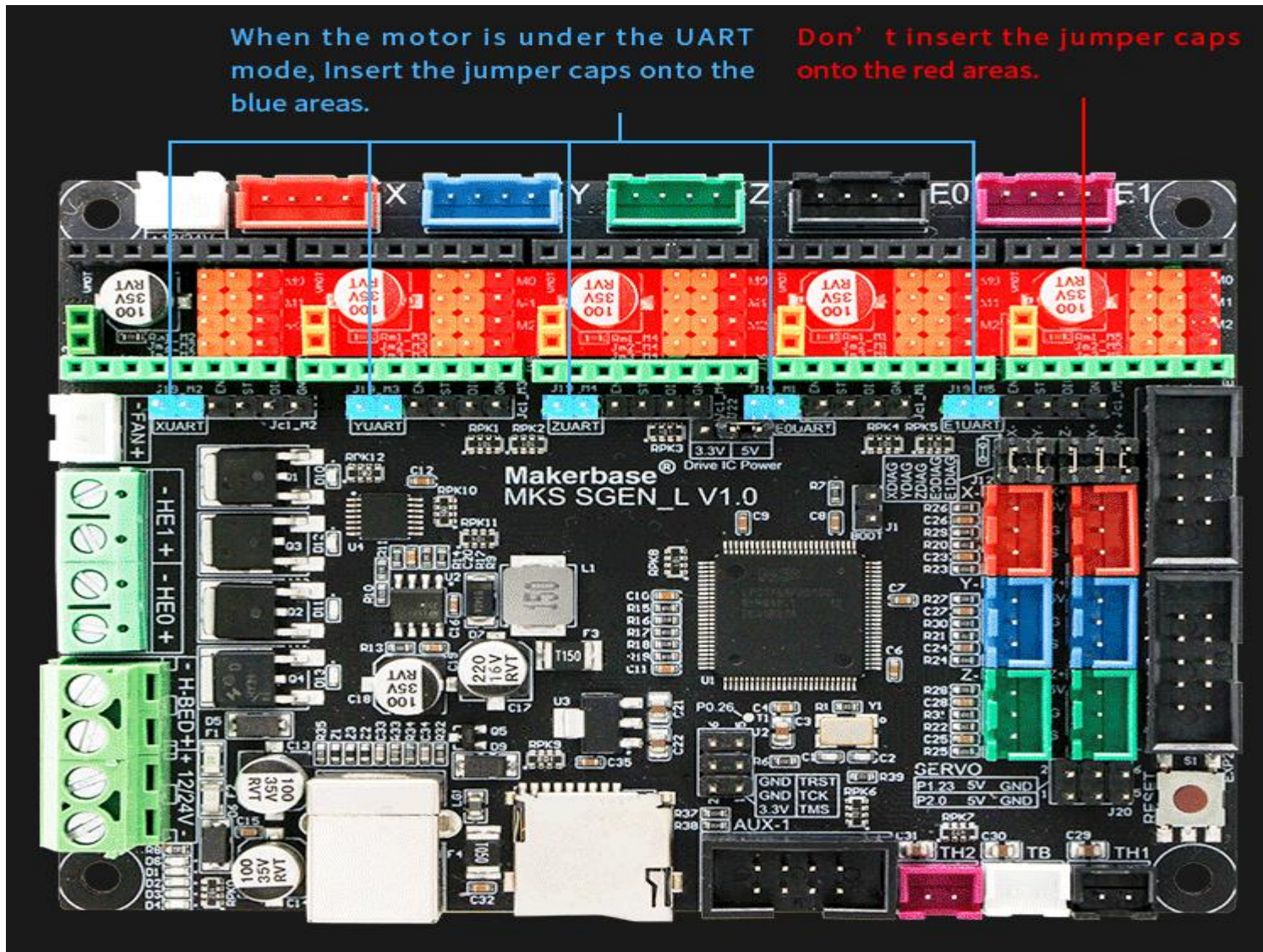
```

Descriptions above are TMC2130 SPI mode setting

UART mode setting (MKS TMC2209, MKS TMC2208 V2.0)

SGEN-L motherboard used with TMC2208V2.0 or TMC2209, requires hardware and software configuration.

The hardware configuration refers to the following picture to set up the jumper caps. If using the TMC2208 V2.0 version, you don't need to connect NC and uart on the driver. However, it doesn't need to connect NC and uart port if using TMC2209



Software Configuration: The configuration method of UART Marlin 2.0 is similar to the 2130.

1. Driver type setting

```

Configuration.h — C:\Users\Administrator\Desktop\MarlinFirmware\Marlin — Atom
File Edit View Selection Find Packages Help PlatformIO

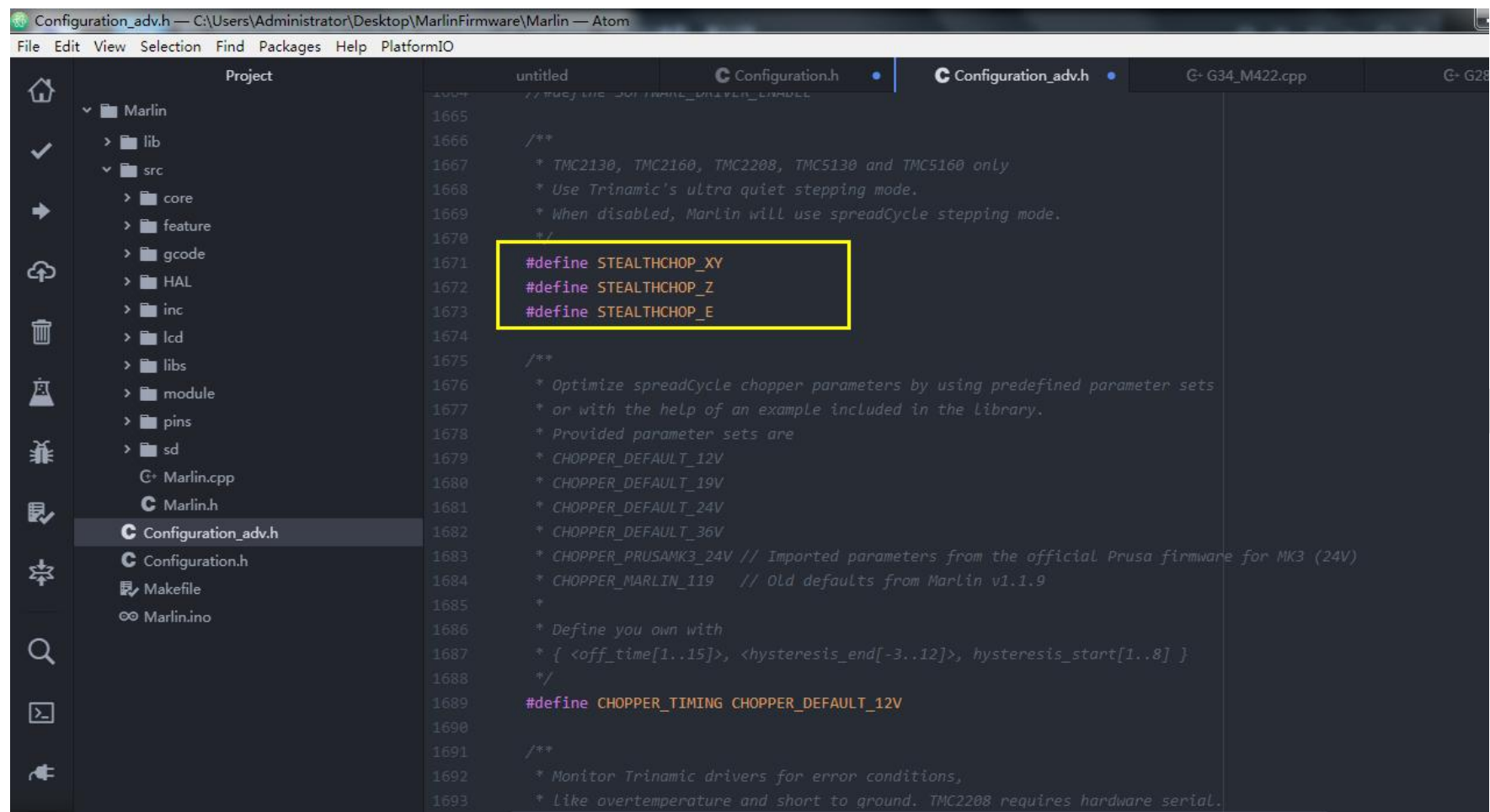
Project: Marlin
  lib
  src
    core
    feature
    gcode
    HAL
    inc
    lcd
    libs
    module
    pins
    sd
  Marlin.cpp
  Marlin.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino

untitled | Configuration.h | Configuration_adv.h | G34_M422.cpp

636 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
637 * stepper drivers that support them. You may also override timing options in Configuration
638 *
639 * A4988 is assumed for unspecified drivers.
640 *
641 * Options: A4988, A5984, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
642 *           TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
643 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
644 *           TMC2160, TMC2160_STANDALONE, TMC5130, TMC5130_STANDALONE,
645 *           TMC5160, TMC5160_STANDALONE
646 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130',
647 *
648 #define X_DRIVER_TYPE  TMC2208
649 #define Y_DRIVER_TYPE  TMC2208
650 #define Z_DRIVER_TYPE  TMC2208
651 // #define X2_DRIVER_TYPE A4988
652 // #define Y2_DRIVER_TYPE A4988
653 // #define Z2_DRIVER_TYPE TMC2130
654 // #define Z3_DRIVER_TYPE A4988
655 #define E0_DRIVER_TYPE TMC2208
656 // #define E1_DRIVER_TYPE A4988
657 // #define E2_DRIVER_TYPE A4988
658 // #define E3_DRIVER_TYPE A4988
659 // #define E4_DRIVER_TYPE A4988
660 // #define E5_DRIVER_TYPE A4988
661
662 // Enable this feature if all enabled endstop pins are interrupt-capable.
663 // This will remove the need to poll the interrupt pins, saving many CPU cycles.
664 // #define ENDSTOP_INTERRUPTS_FEATURE
    
```

2. Operation mode selection

It will be in Silent drive mode when STEALTHCHOP is enabled. If STEALTHCHOP is commented out (add : “//” before the item)), it will be in SpreadCycle.



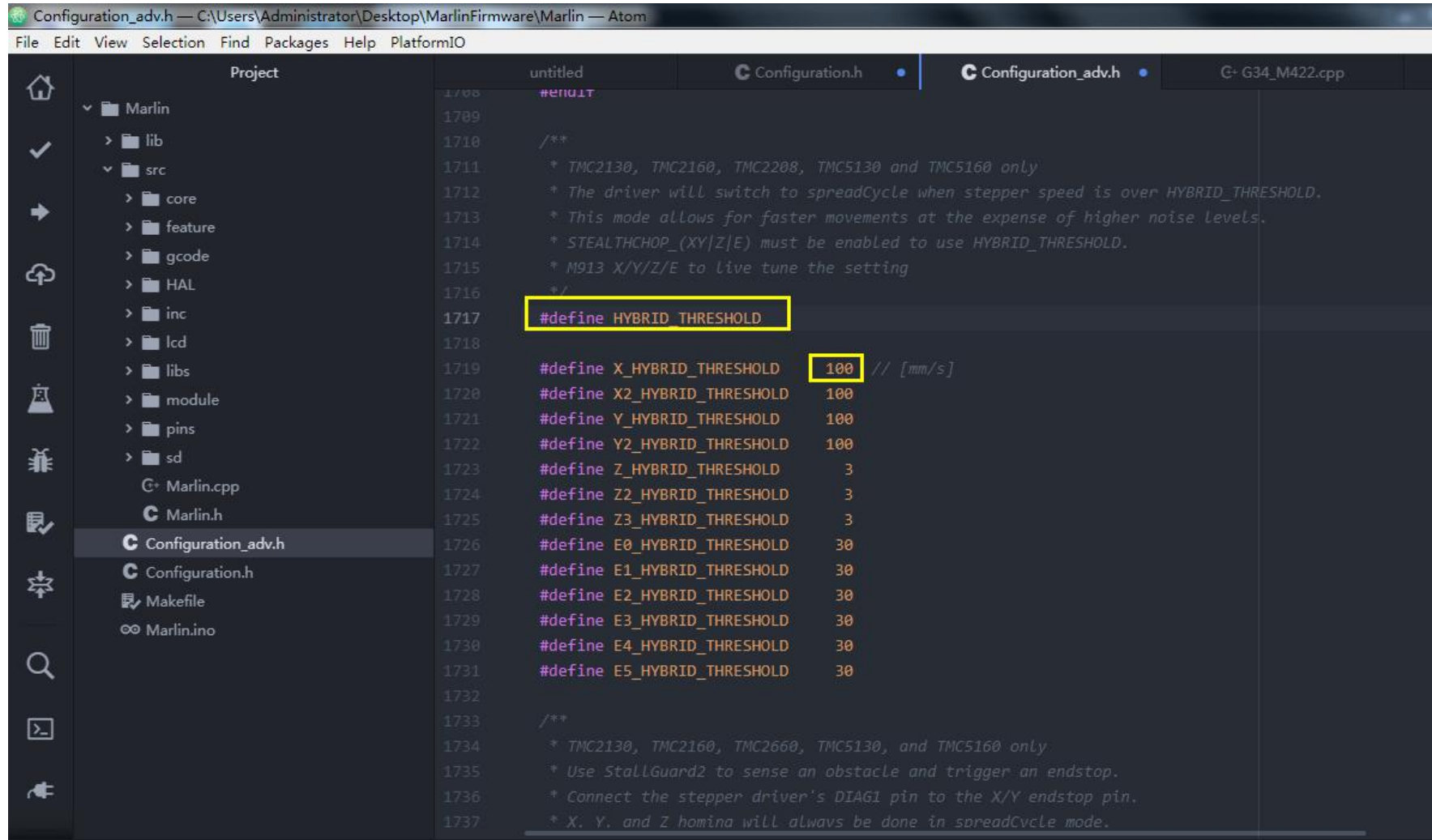
```
1665 // TMC2130, TMC2160, TMC2208, TMC5130 and TMC5160 only
1666 /**
1667  * TMC2130, TMC2160, TMC2208, TMC5130 and TMC5160 only
1668  * Use Trinamic's ultra quiet stepping mode.
1669  * When disabled, Marlin will use spreadCycle stepping mode.
1670  */
1671 #define STEALTHCHOP_XY
1672 #define STEALTHCHOP_Z
1673 #define STEALTHCHOP_E
1674
1675 /**
1676  * Optimize spreadCycle chopper parameters by using predefined parameter sets
1677  * or with the help of an example included in the library.
1678  * Provided parameter sets are
1679  * CHOPPER_DEFAULT_12V
1680  * CHOPPER_DEFAULT_19V
1681  * CHOPPER_DEFAULT_24V
1682  * CHOPPER_DEFAULT_36V
1683  * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
1684  * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
1685  *
1686  * Define your own with
1687  * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
1688  */
1689 #define CHOPPER_TIMING CHOPPER_DEFAULT_12V
1690
1691 /**
1692  * Monitor Trinamic drivers for error conditions,
1693  * like overtemperature and short to ground. TMC2208 requires hardware serial.
```

If you want to configure the hybrid mode, you must first enable the STEALTHCHOP mode.

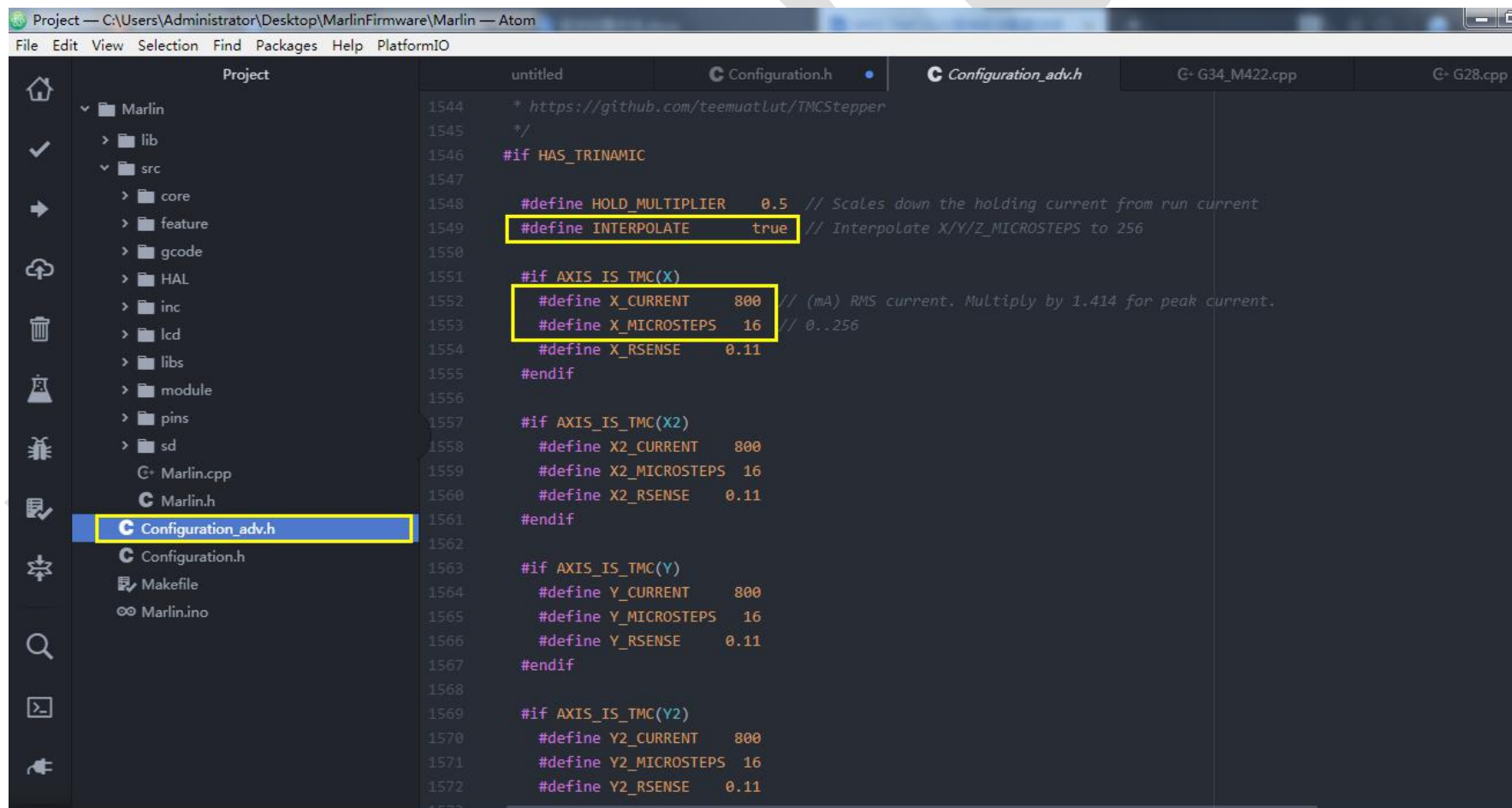
Enable #define HYBRID_THRESHOLD

The value corresponding to the HYBRID_THRESHOLD item is the value of the mode switch. If the motor runs over 100mm/s faster than the setting value, it will be switched to the spreadcycle mode automatically, and below 100mm/s it will be the stealthchop mode.

Mixed mode can be selectively configured according to your own needs



1. Driver current setting



Microstep setting: X_MICROSTEPS. Generally, it will be set as 16 microsteps, because enable `#define INTERPOLATE true`, chip will internally expand it to 256 microsteps.

The real current value of driver is 1.414 times as the setting one. Thus, it is not recommend to set the current over the maximum current of driver or motor.

4. UART (RX 与 TX) pin setting

Because the wires are internally integrated to motherboard, pcb file “pin” setting has been finished., which needn't to be modified.

That is all contents about the driver setting.

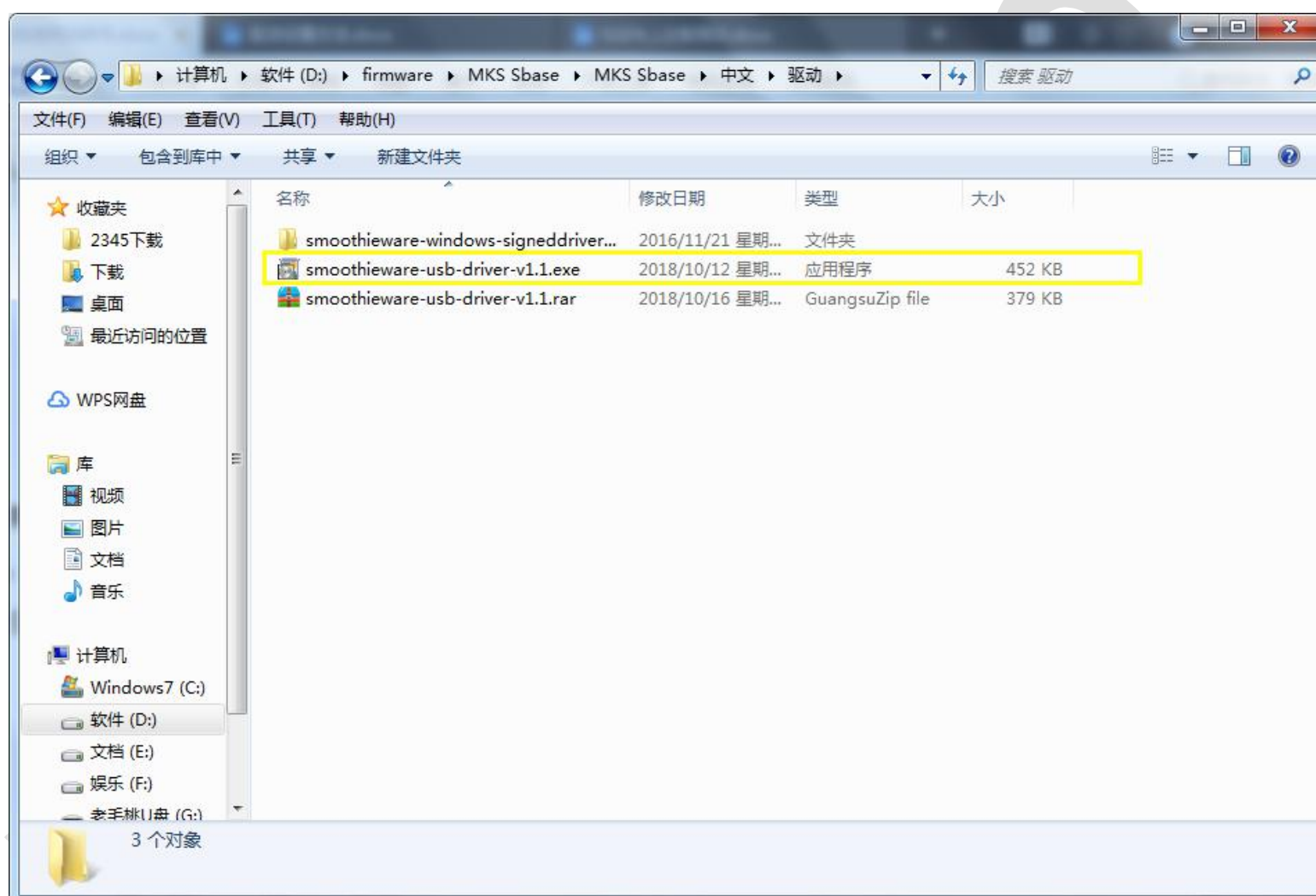
Makerbase

VI Printing connection and driver installation

If it needs to connect computer to print, the computer needs to be installed a USB driver to recognized the COM port.

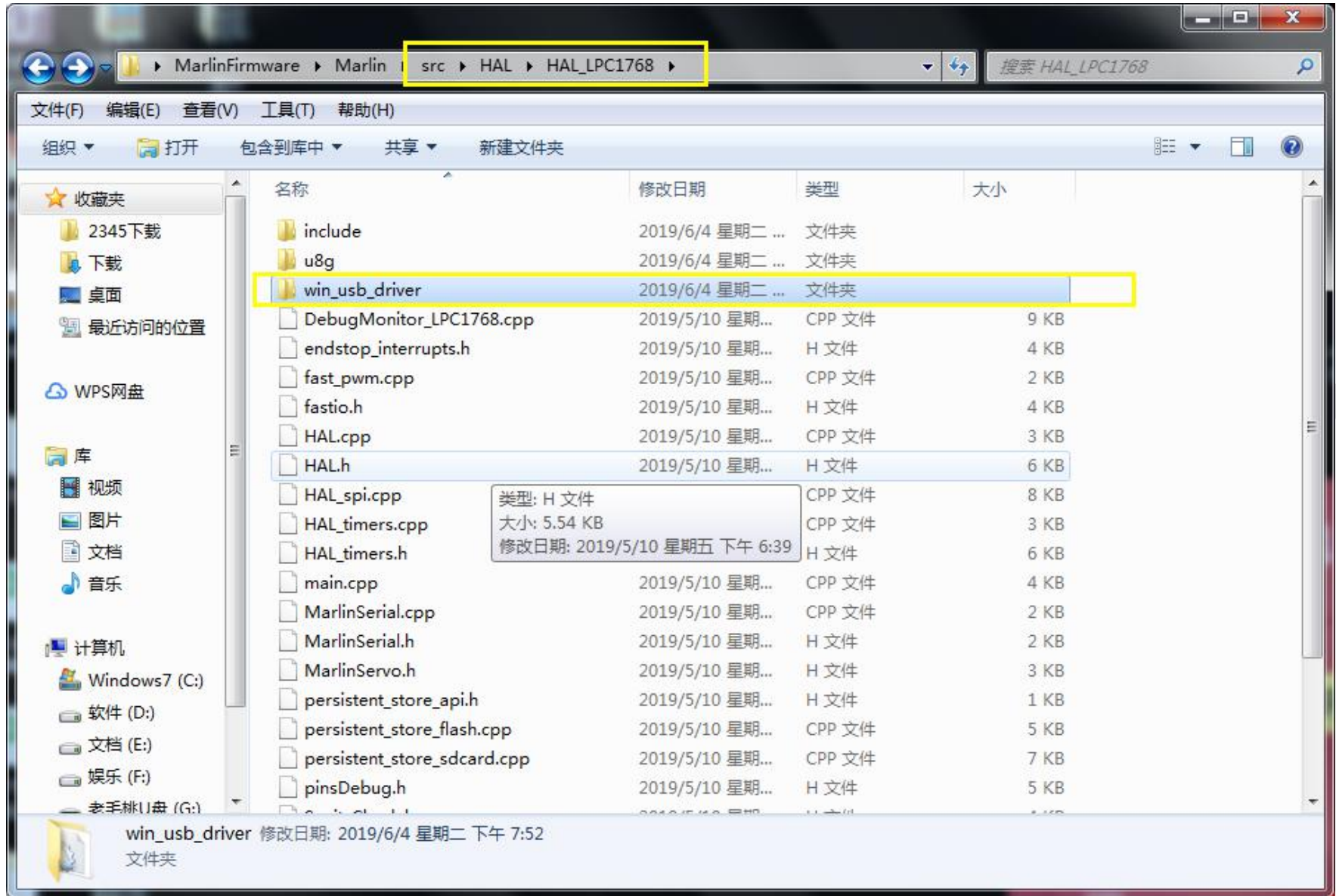
Note: only update the motherboard firmware, can motherboard COM port be recognized by computer.

The USB driver of Smoothieware can be download and install via official website, or get zipped files from customer service staffs and technicians.



MARLIN2.0 firmware

If the marlin firmware is updated, but the computer recognizes it as an unknown device, you can right-click to select the update driver and select the path to update the driver.



VII Technical support and guarantee

1. All our products have a power-on test before delivery to ensure its normal use.

2. Welcome to join Blog: <http://flyway97.blog.163.com>

3. Custom 3D printer motherboard, please contact:

Mr.Huang:13148932315

Mr.Tan:15521395023

Mr.Peng: 13427595835

E-MAIL:caixindong@makerbase.com.cn

4. Any problems, Please contact our customer service staffs and technicians. We are glad to help you.



Official website