



Makerbase

广州谦辉信息科技有限公司

MKS SGEN-L 主板使用说明

创客基地

QQ 群：489095605 232237692

邮箱：Huangkaida@makerbase.com.cn

文档版本 1.0

发布日期 2019-6-11

权所有 © 广州谦辉信息科技有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

和其他  Makerbase 或” Makerbase “商标均为广州谦辉信息科技有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受广州谦辉信息科技商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，广州谦辉信息科技有限公司对本文档内容不做任何明示或默示的声明或保证。由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

版本更新

版本	修改时间	修改内容	备注
V1.0	2019.06.11	1. 初始版本	

Makerbase

目录

一、简介.....	5
二、特点优势.....	6
三、连接说明及尺寸图.....	7
1、MKS SGen-L 主板正面实物图.....	7
2、MKS SGen_L 主板系统连接图.....	7
3、MKS SGen_L 主板安装尺寸图.....	8
4、MKS SGen 主板 PIN 口图.....	8
四、使用说明.....	9
1、 固件获取方式:	9
2、 更新固件的方法.....	9
2.1 smoothieware 固件更新.....	9
2.2. Marlin2.0 固件更新.....	10
五、 参数配置说明.....	16
5.1-smoothieware 参数配置.....	16
5.2 marlin2.0 参数配置.....	22
5.2.1 基本参数配置.....	22
5.2.2 自动调平设置 (z-probe 和 3d touch)	32
5.2.3 驱动模式设置.....	37
普通模式 (step/dir 模式)	37
TMC2130 SPI 模式设置.....	38
TMC2208 UART 模式设置.....	43
六. 连接打印与驱动安装.....	47
七. 技术支持及保证.....	49

一、简介

MKS-SGen_L 是创客基地推出功能非常强大的 3D 打印机 ARM 主板，采用 32 位的 Cortex-M3 的 LPC 1768 处理器，主频高达 100MHZ，运算处理能力十分稳定。可兼容 Smoothieware 和 marlin2.0 固件，适应多样需求。拓展性强大，能和多种驱动配套使用。可以支持外接和直插的方式，直插驱动兼容各种驱动的特殊模式（uart，spi 等模式。）兼容多款 lcd 屏幕和 MKS 系列触摸屏。是一款功能多样且性价比极高的主板。

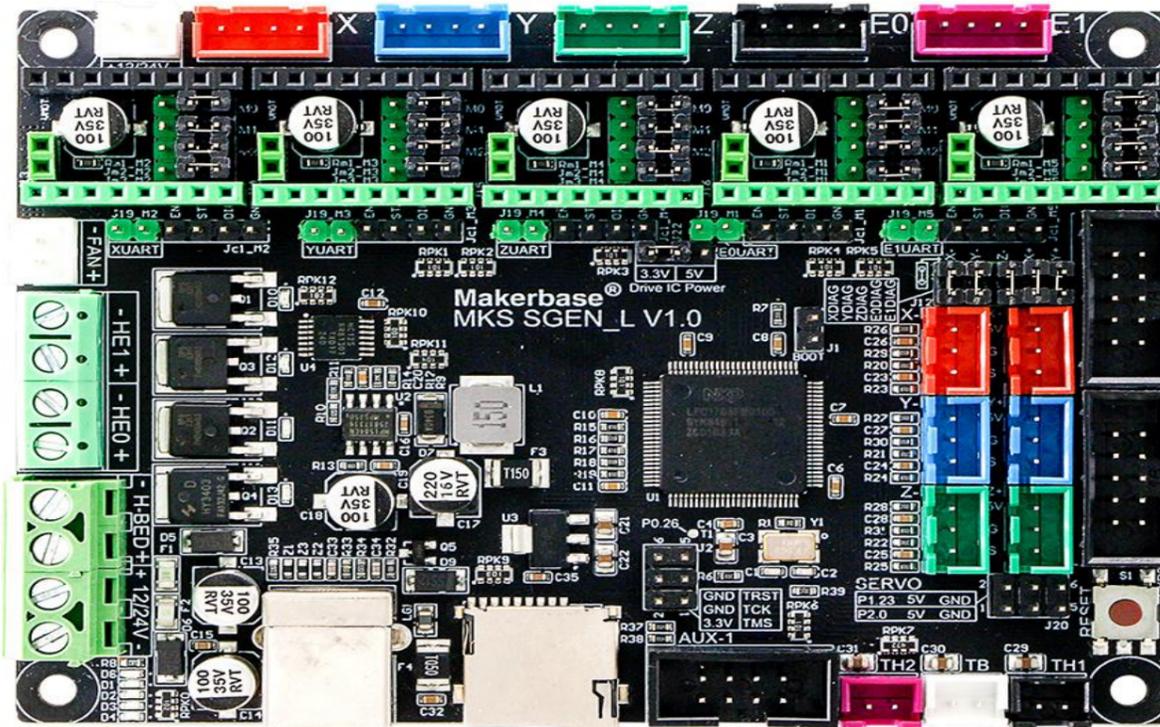


二、特点优势

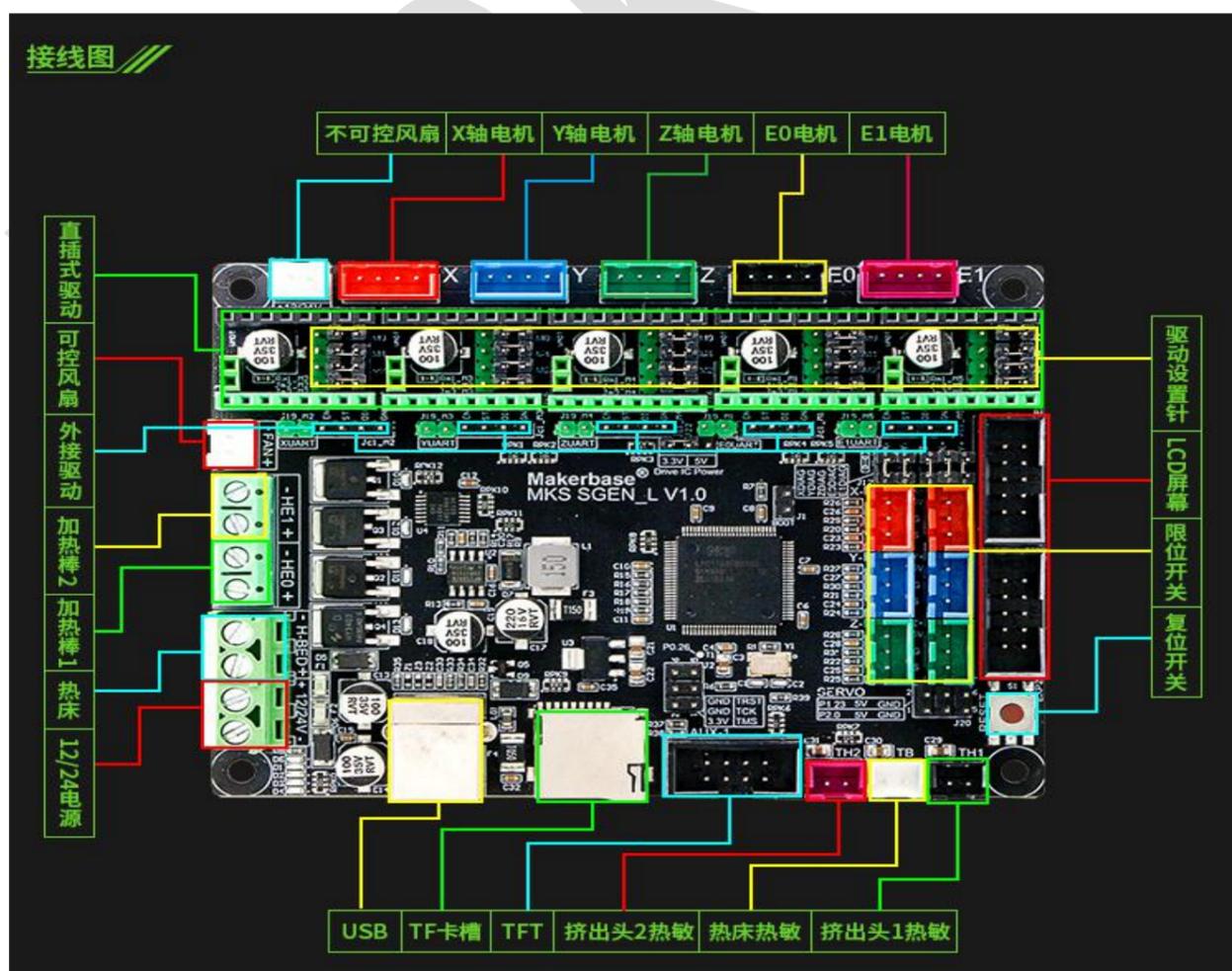
- ◆ 采用 32 位 Cortex-M3 的 LPC 1768 主控芯片，100MHZ 主频，运算能力强，性能大幅提升。
- ◆ 兼具直插式驱动插口与外接驱动端口，拓展性更强，选择性更多。
- ◆ 兼容多种驱动和特殊模式（TMC2208 的 uart 模式，TMC2130 的 SPI 模式等）有专门的端口的预留，接线设置更简洁方便。
- ◆ 支持多种开源固件，延续原来的 s 系列主板采用的 smoothieware 固件同时，SGen-L 也可以支持 marlin2.0 版本的固件
- ◆ 采用高质量线路板，且各个接口的对应的 pin，都有丝印进行标注，更容易识别修改。
- ◆ 采用专用电源芯片，支持 12V-24V 电源输入。
- ◆ 支持创客基地开发的 MKS LCD12864, MKS MINI12864 等控制板进行显示控制。
- ◆ 可与 MKS TFT 全系列的触摸屏配合使用。

三、连接说明及尺寸图

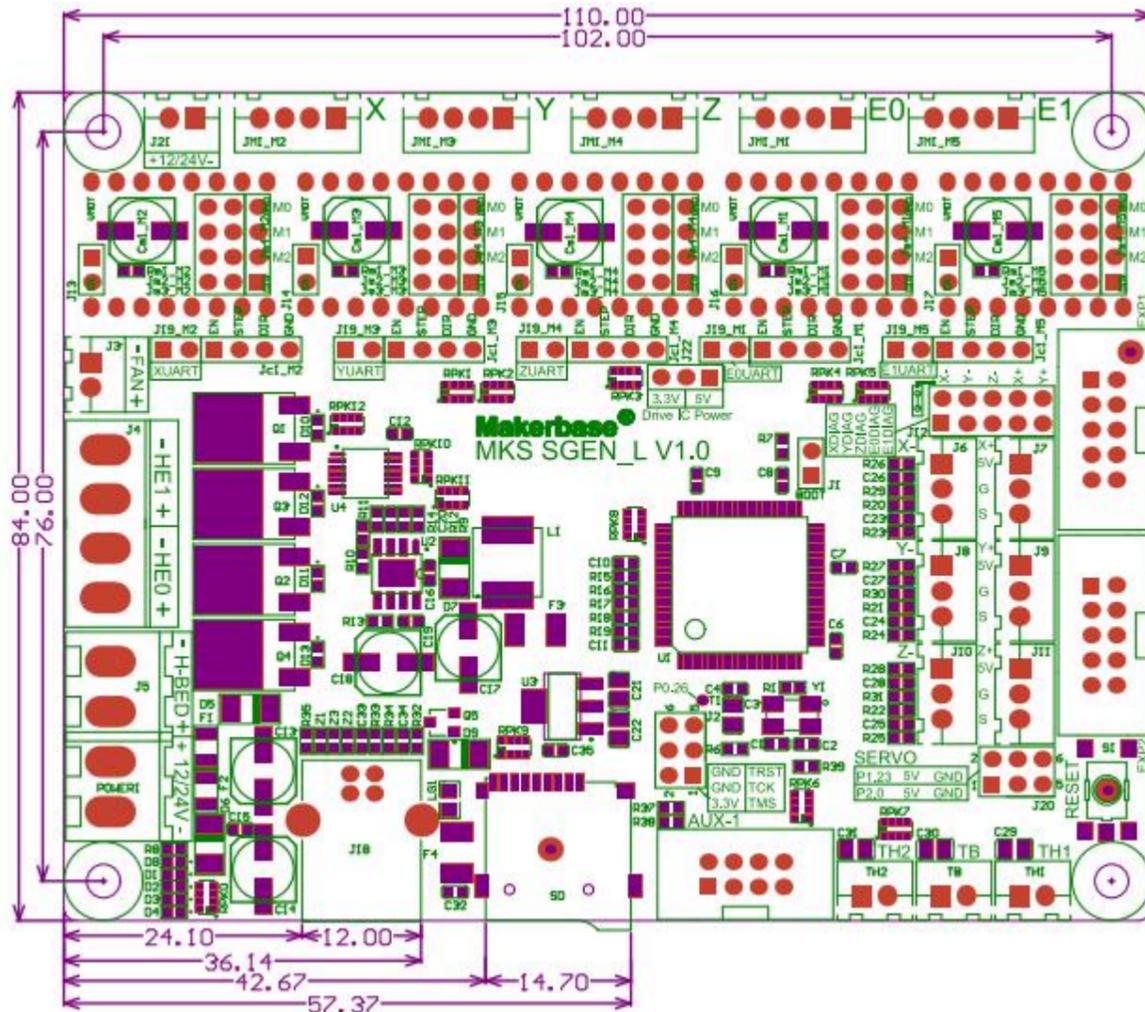
1、MKS SGen-L 主板正面实物图



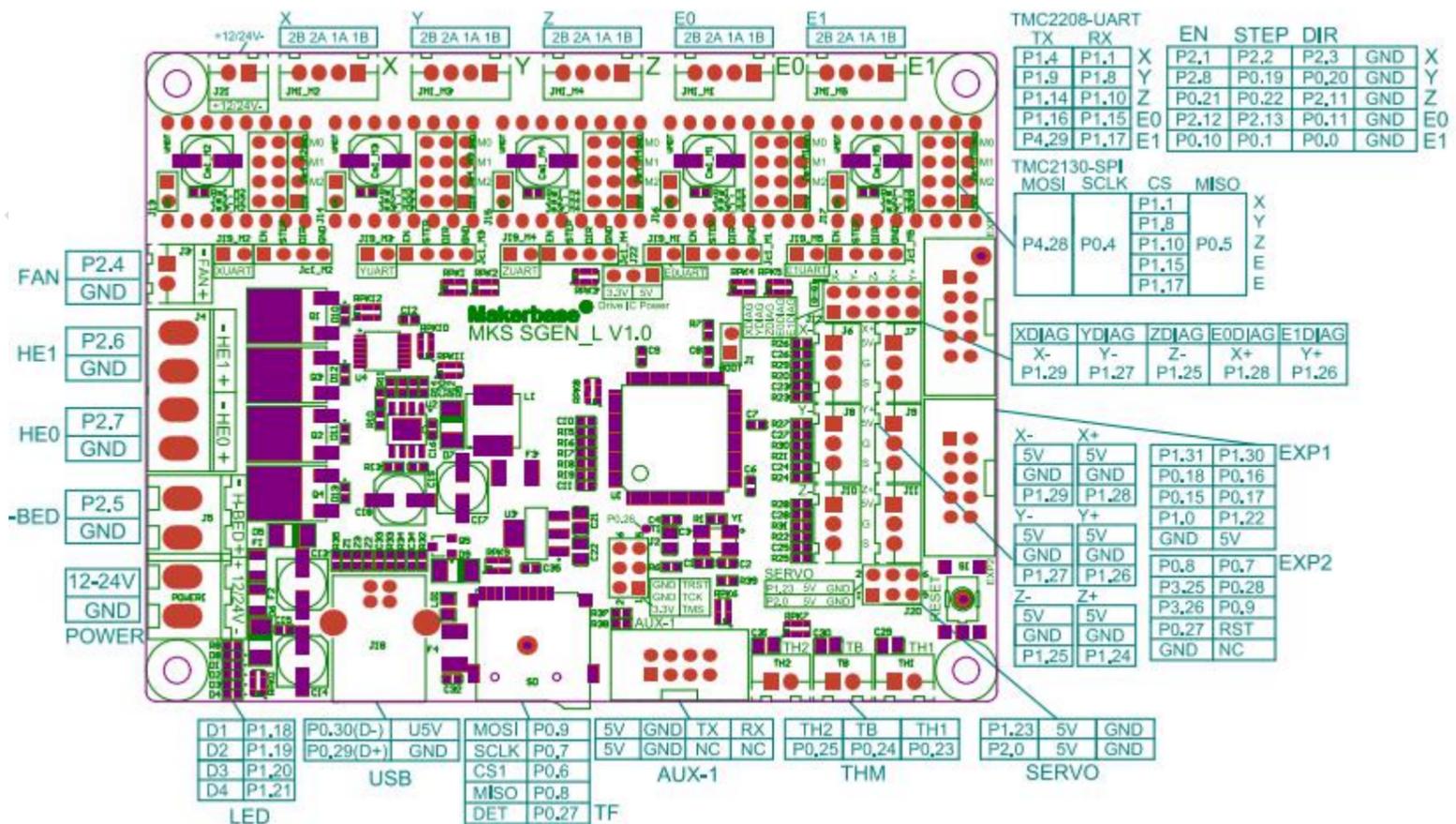
2、MKS SGen_L 主板系统连接图



3、MKS SGen_L 主板安装尺寸图



4、MKS SGen 主板 PIN 口图



四、使用说明

1、 固件获取方式：

- 1.1 问淘宝客服或者技术人员获取固件；
- 1.2 在创客基地讨论群的群文件中下载；
- 1.3 登录网址下载进行下载：<https://github.com/makerbase-mks?tab=repositories>
- 1.4 前往百度网盘进行下载：<https://pan.baidu.com/s/170g2R8-s7tnNEL1GpoB-vA>
提取码：g6gn

2、 更新固件的方法

其中包括 smoothieware 固件的更新与 marlin 固件的更新

2.1 smoothieware 固件更新

- 2.1.1 将更新程序拷贝到 TF 卡根目录，包括：1、固件 firmware.bin 2、配置文件 config.txt
- 2.1.3 将 TF 卡插入卡槽中并重新上电，系统会自动升级新固件，升级成功后，SD 卡中的文件被改名为 firmware.cur。
- 2.1.3 等主板更新完固件之后一会儿，主板左上角指示灯正常为 D1 D4 D7 为常亮，D2 D3 闪烁；

2.2. Marlin2.0 固件更新

Marlin2.0 固件的更新的方法类似，将 firmware.bin 文件拷到 TF 卡中，主板上电更新即可。

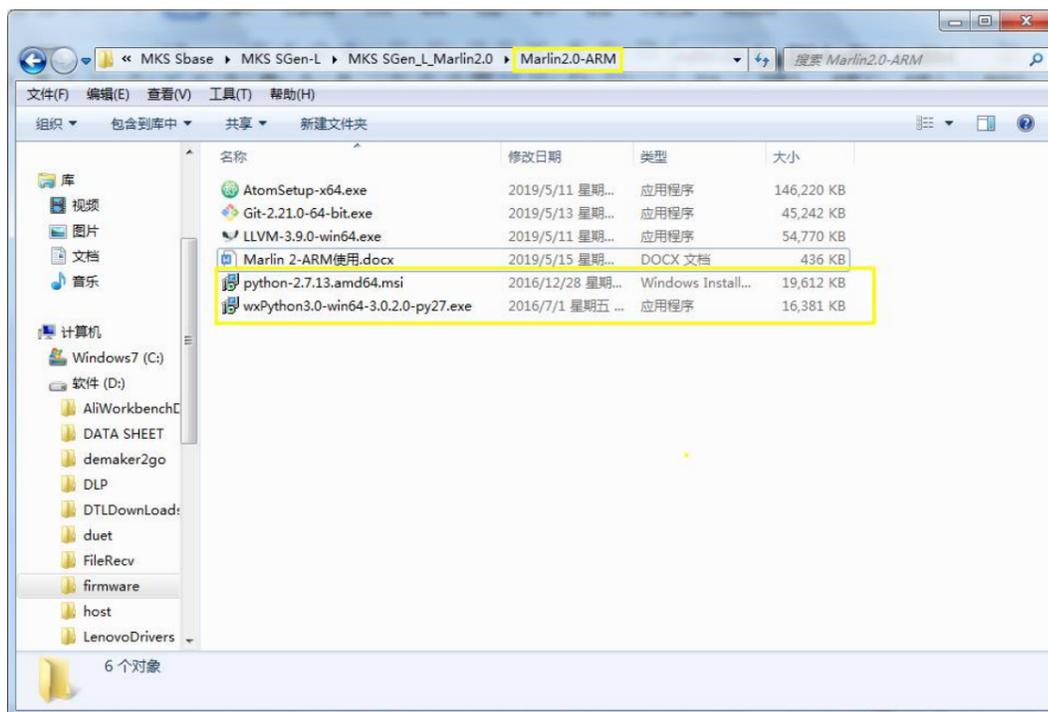
更新成功文件将会变成 firmware.CUR 文件

Marlin2.0 固件不能像 smoothieware 一样，通过 config 文件进行修改配置。Marlin2.0 每次更新参数配置都需要生成新的 firmware.bin 文件进行重新更新。

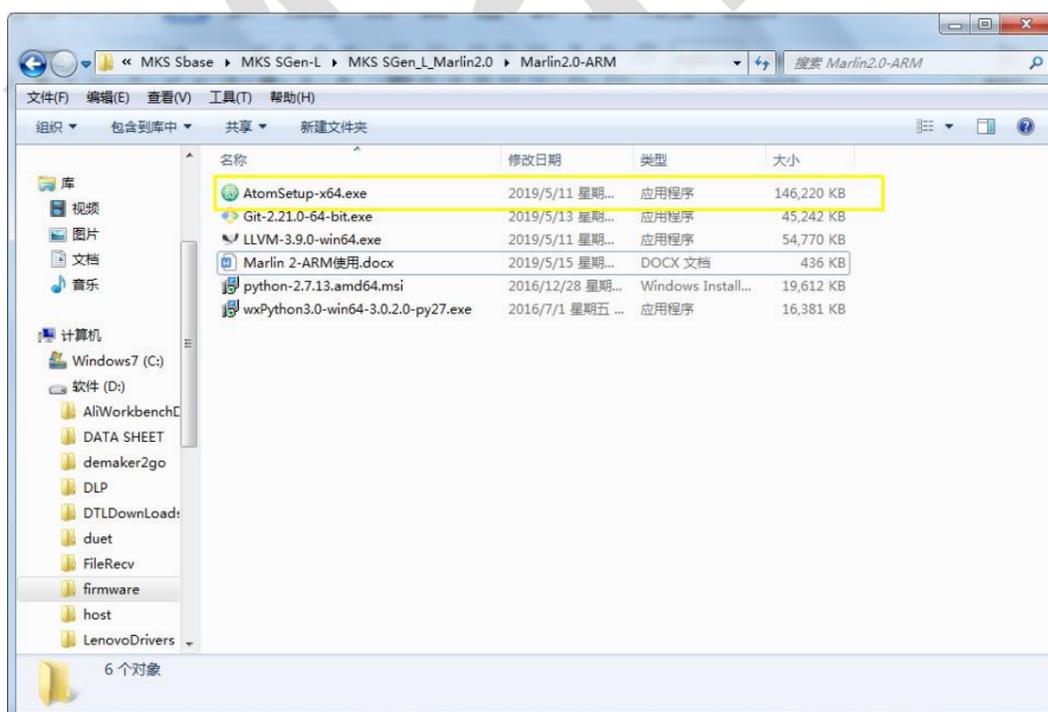
2.2.1 编译环境的搭建和 bin 文件的生成

1. 下载指定网址下载固件和软件的压缩包

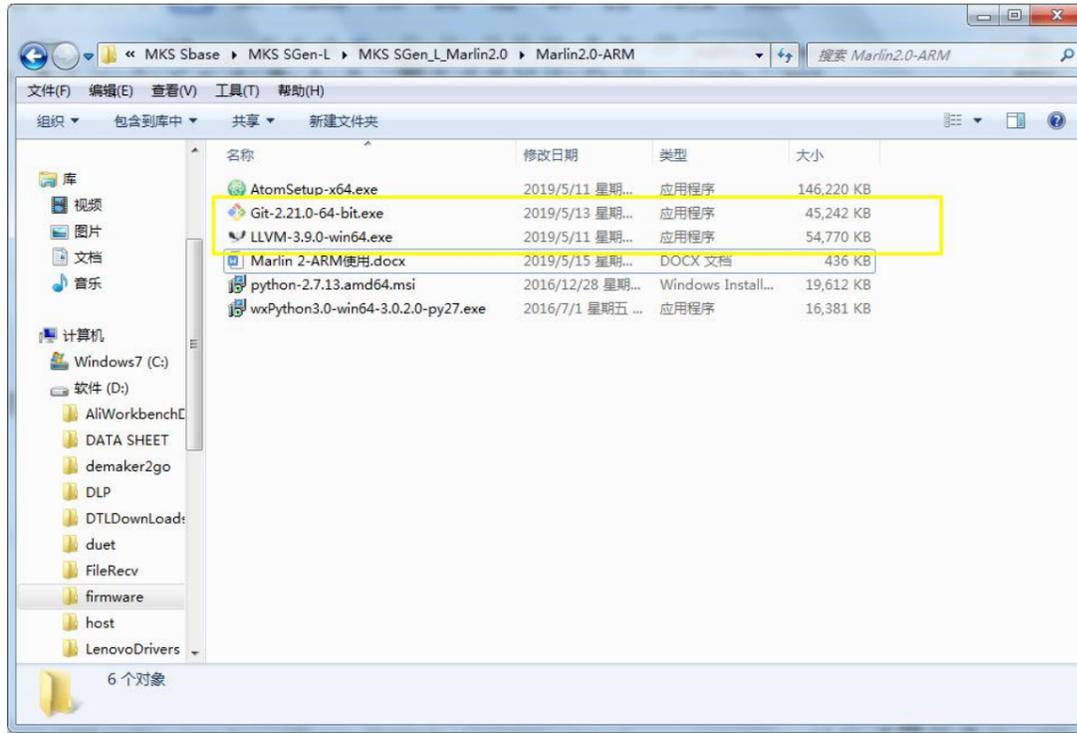
2. 打开 marlin2.0-ARM 文件夹，安装 python-2.7 和 wxPython3.0, 点击安装包安装，如图所示



3. 编译平台 Atom 安装



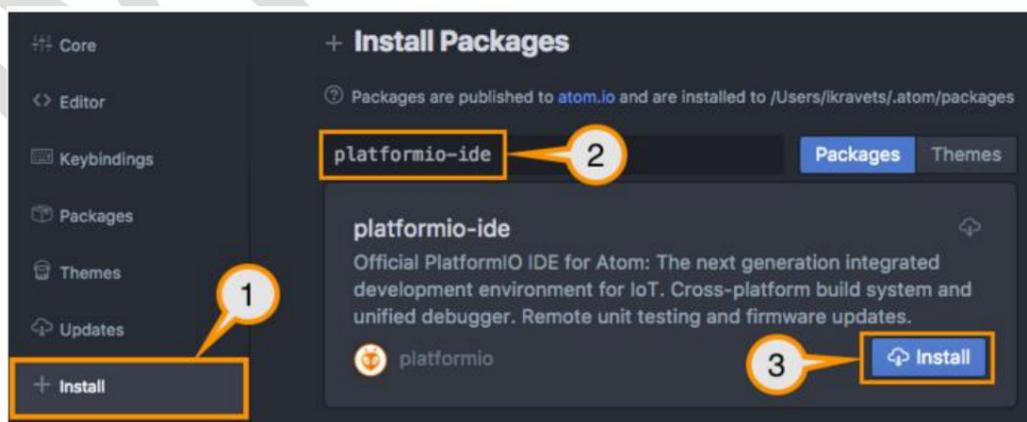
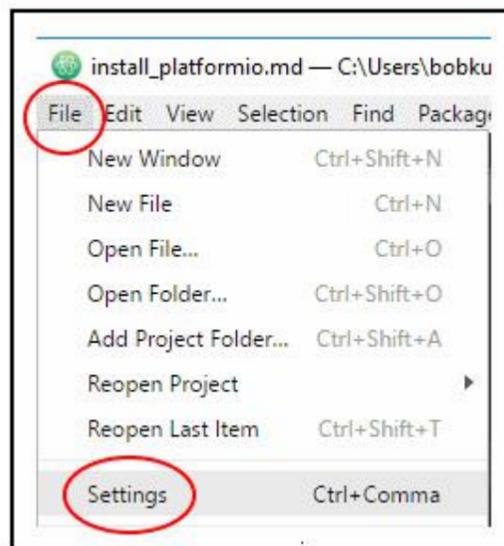
4. 安装 Git-2.21.0 和 LLVM（只支持 3.9.0）

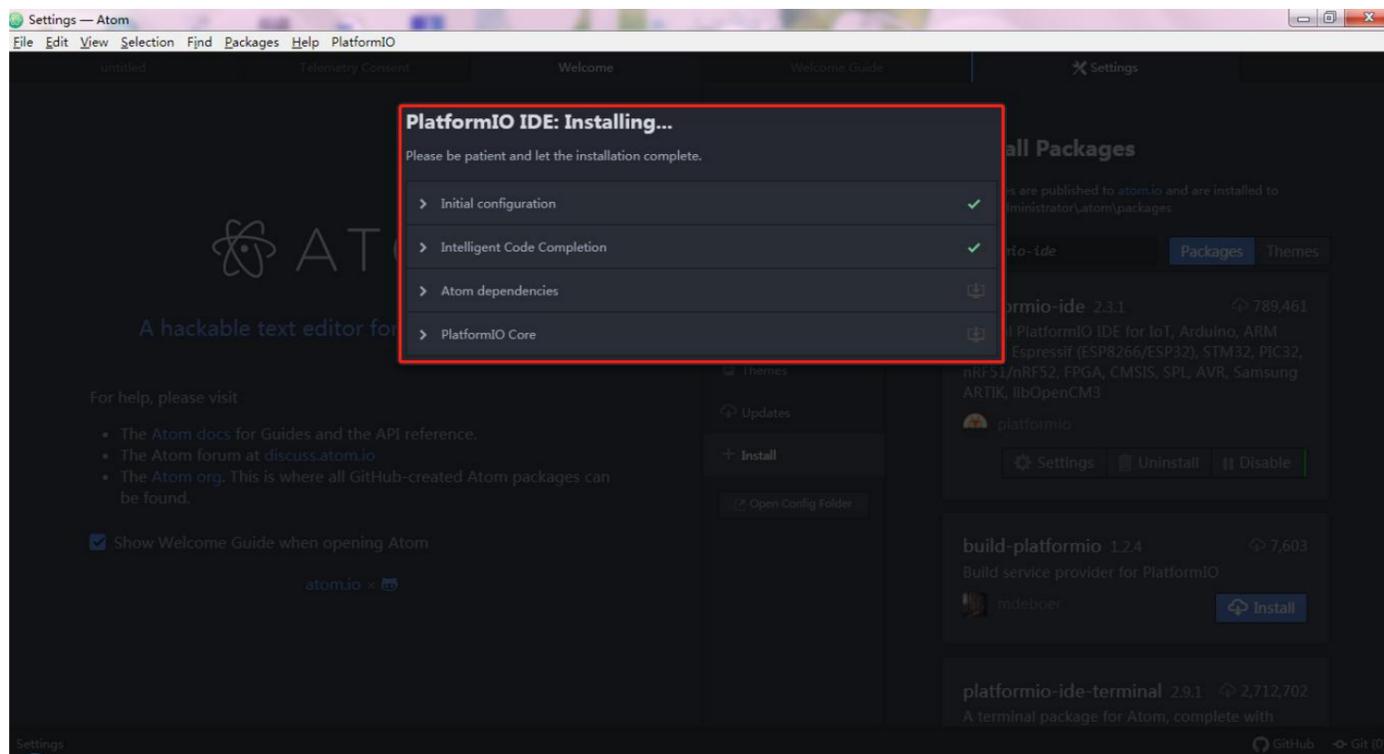


5. Atom 软件插件安装

安装好 Atom 后打开 Atom,打开 File→Settings→install, 需安装两个插件
搜索 platformio_ide 安装。

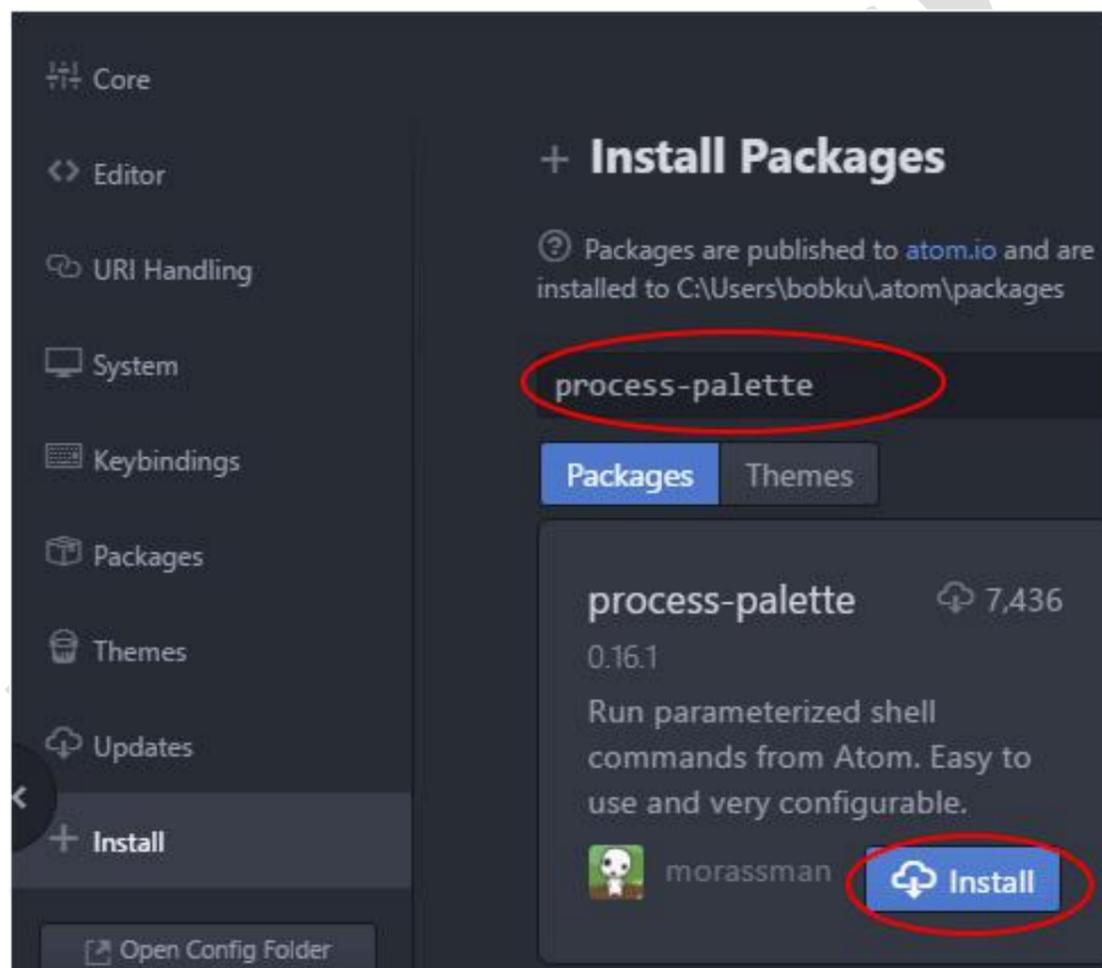
- Windows: Click File then click Settings





等待安装完成即可

搜索 process-palette 安装。



安装完成之后，即可导入 marlin2.0 源码进行修改编译。

以上为操作之后，则编译环境搭建完成

6. Bin 文件编译

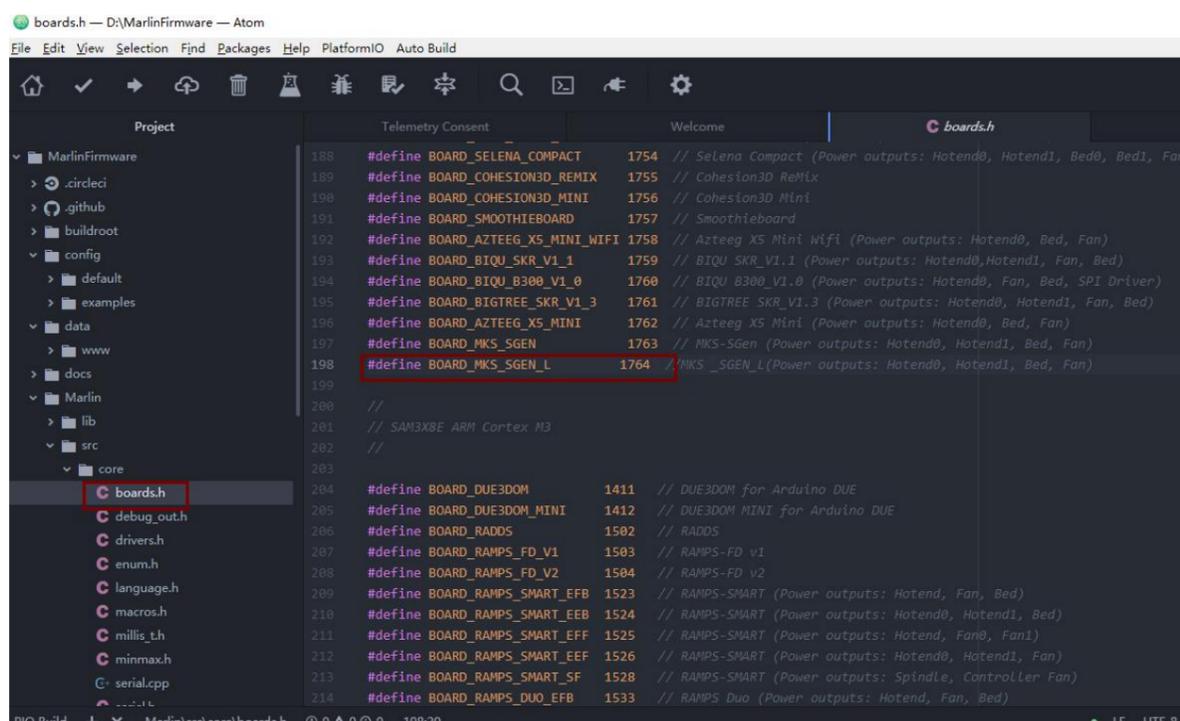
1、固件工程的导入

(Marlin2.0 固件的文件路径不能有中文建议路径不要太长，否则编译出错)点击 File→Open Folder,打开 marlin2.0 固件的文件夹导入 marlin 固件。

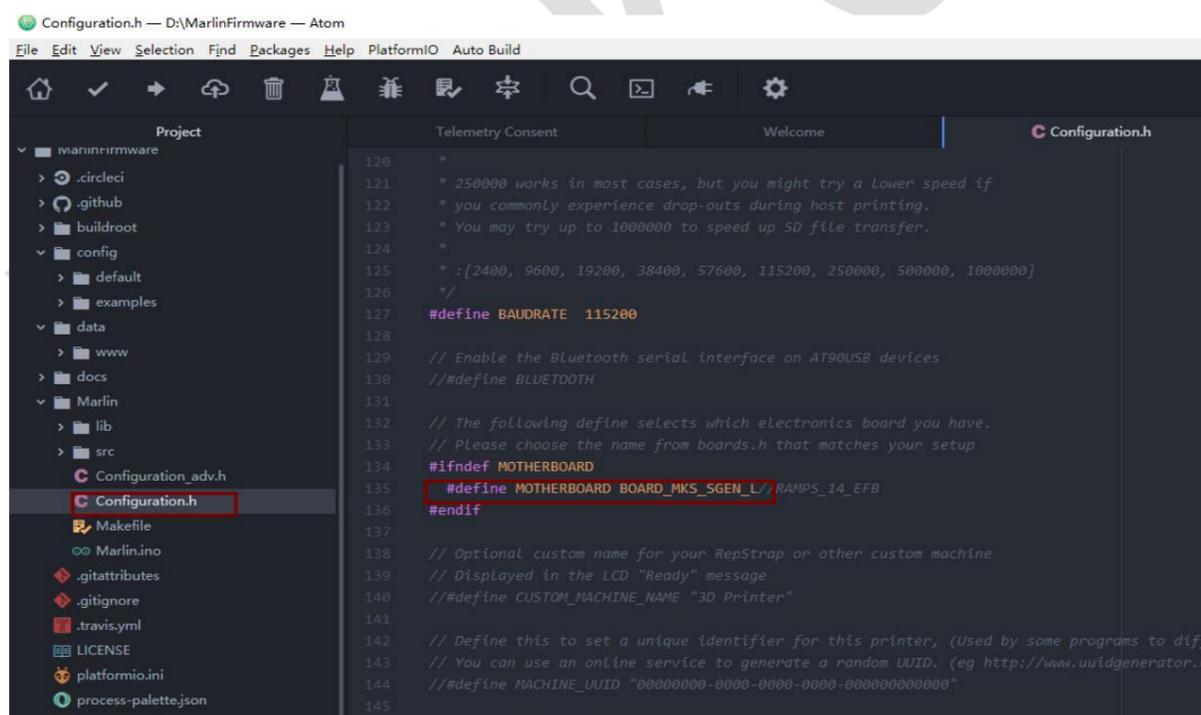
2、主板型号定义

打开 board.h，能找到 MKS_SGEN_L 的主板类型定义为 BOARD_MKS_SGEN_L

需下载 github 和网盘指定 marlin2.0 文件才有 SGEN_L 的主板类型，其他公版 marlin2.0 暂时没有这个主板类型。后续会在公版进行添加



在配置文件 configuration.h 中将主板类型配置为 BOARD_MKS_SGEN_L

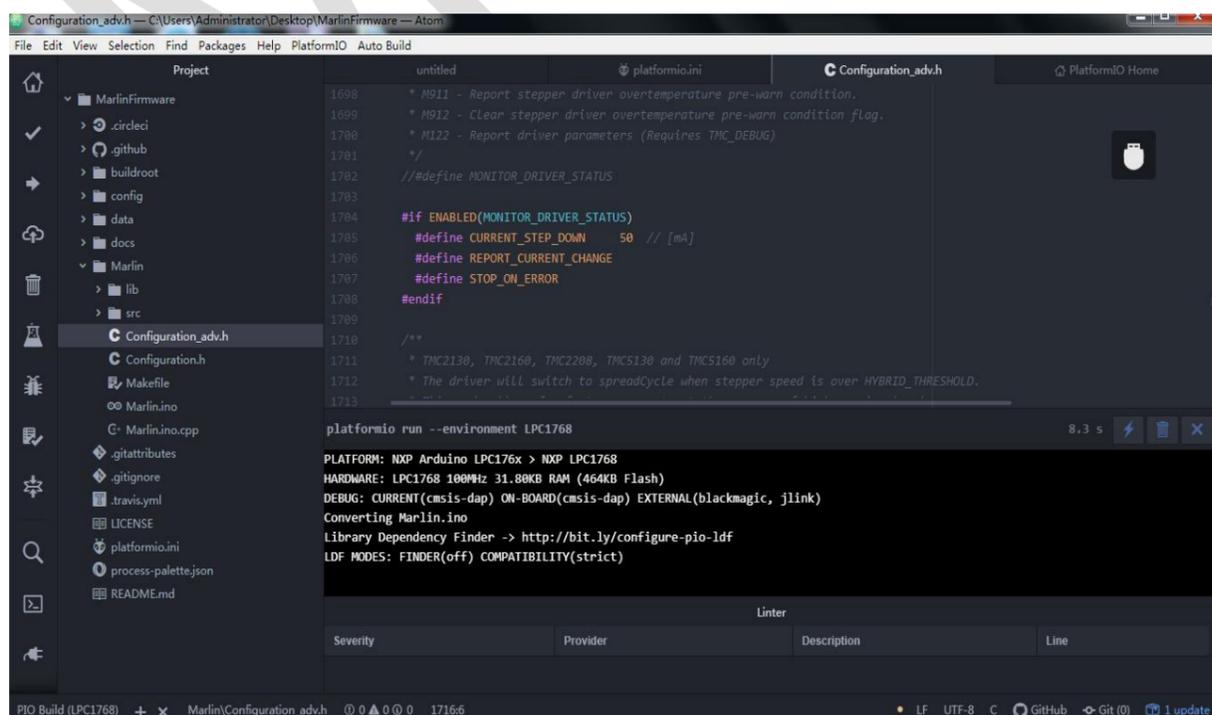
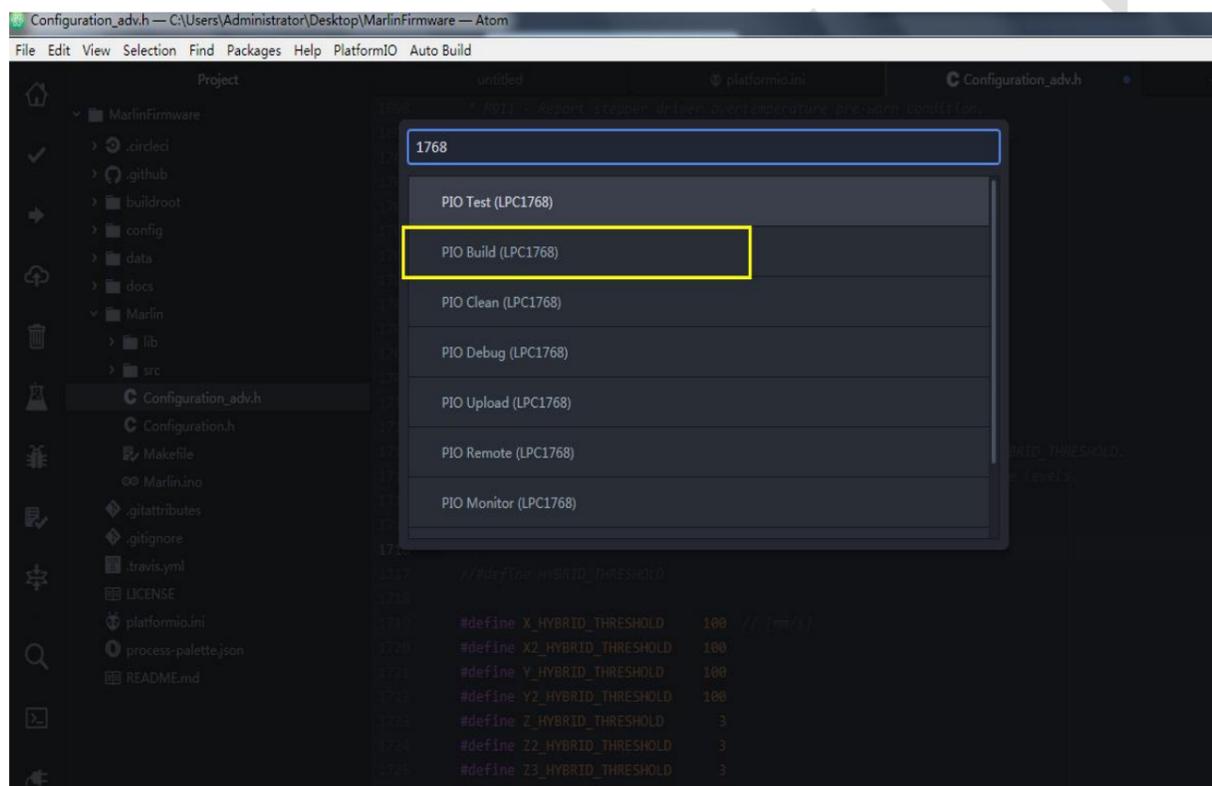
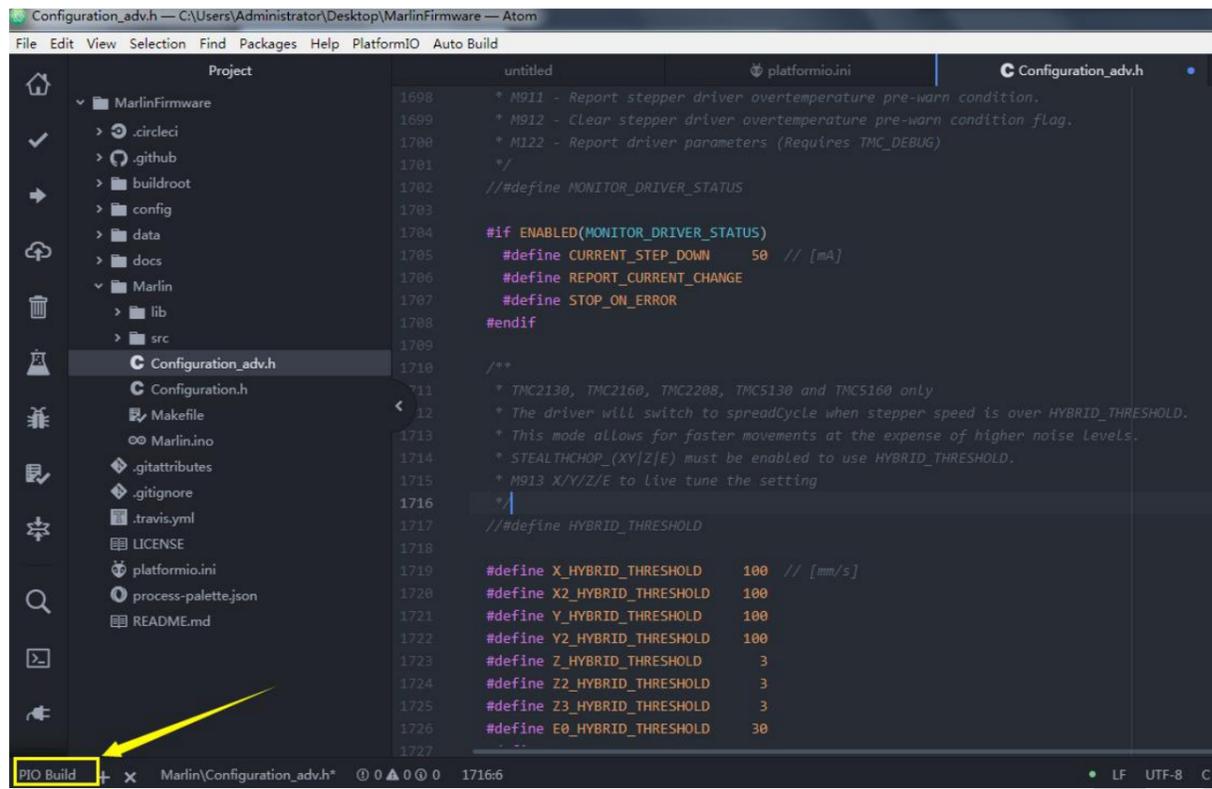


3. 根据机器情况进行参数配置（详情见 marlin2.0 基本参数配置）

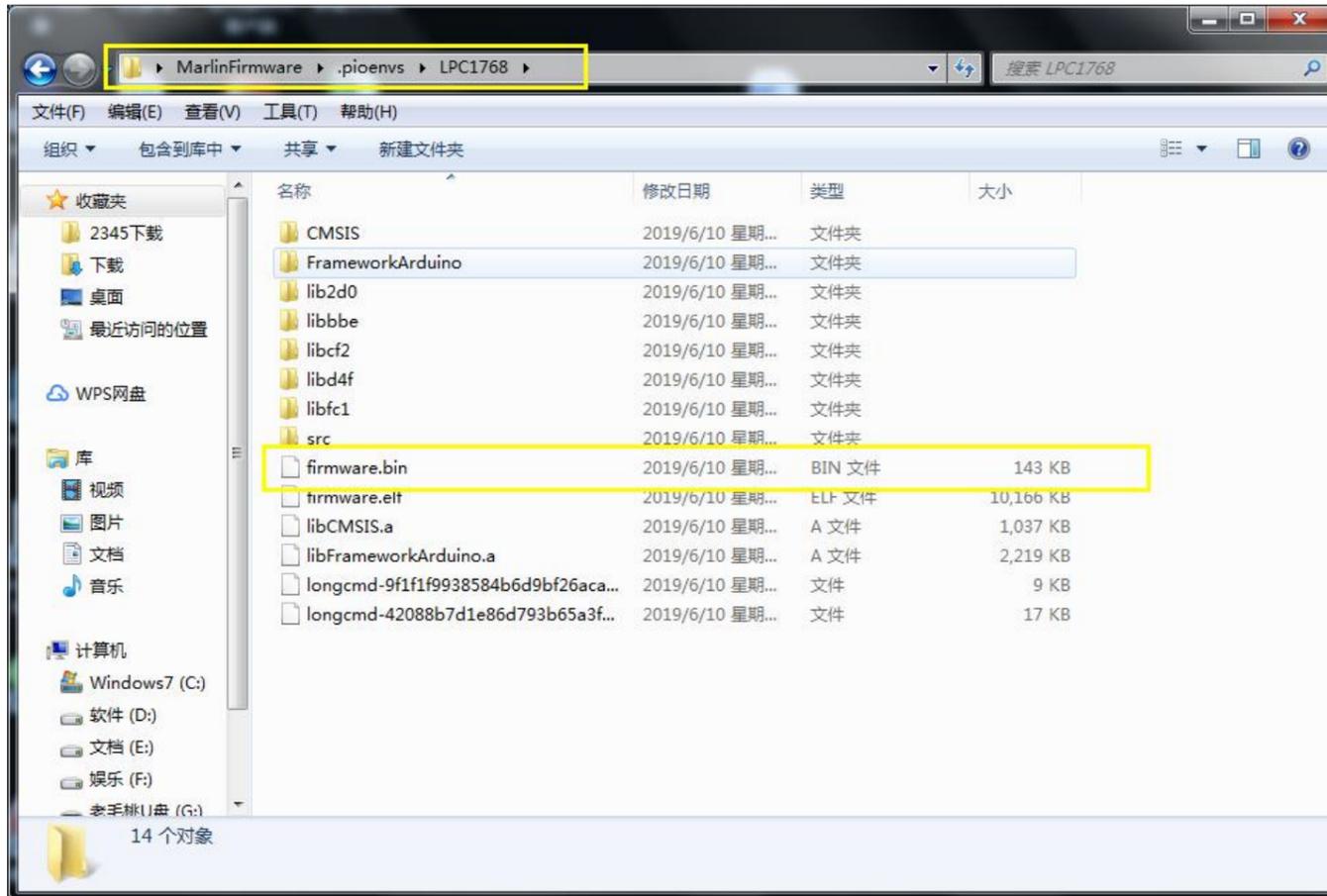
4. 固件编译

点击左下角的 PIO Build → 输入框输入芯片型号 1768 → PIO Clean(LPC1768) → PIO Build(LPC1768),编译完后打开 marlin 固件的文件夹 → 打开 .pioenvs → 复制 firmware.bin 到 TF 卡

插到主板卡槽刷固件。



编译完成之后在这个路径下会有一个 firmware.bin 文件，将文件拷打 TF 卡根目录更新即可。



五、参数配置说明

5.1 smoothieware 参数配置

1. smoothieware. 基本参数配置

(1) 以下是需要根据机器修改的参数

直接修改 SD 卡上的 config.txt 文件，保存后重新上电，参数配置即可生效。

```
#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
..
```

注：因为电机驱动改为直插式驱动，所以驱动电流的调节，不能在配置文件里面进行调节。而是通过直插驱动上的旋钮进行调节。(详情参考下文，驱动电流的调节和注意事项)。

(2) 以下是根据需要进行选择的参数

```
#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
```

各个轴运行的速度和加速度设置，可根据自身需求设置。

如果出现回零状态异常，则可以对相应轴的开关的开关类型进行修改。

(3) 挤出机 E0 参数设置

#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           #
  
```

默认设置即可

(4) 双挤出头参数设置

如果是使用双头打印，需要去掉以下配置前面的注释（#）。

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      #
  
```

对应的 pin 脚和类型信息，在不是很熟悉的情况下，建议使用以上默认设置。

(5) 热床设置

```

#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
  
```

上图为使能热床，如果不需要用到则改成 false 则为禁用热床

以上两部分加热头和热床的默认传感器类型 temperature_control.module_name.thermistor RRRF100K

为普通的 100K NTC 热敏电阻。如果使用其他类型的传感器，则需要到固件官网，查询对应型号所对应的名称。

.PID 设置

用于调节关于固件参数问题造成的温度跳动过大的问题

首先运行 M303 指令，例如：

```
M303 E0 S190
```

针对打印头 E0 经常加热到 190 度的情况，自动运行 PID 运算

#系统运行大约 8 个循环，显示类似以下信息

```

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
  
```

可以把得到的 PID 值直接写入 config.txt，或者直接运行 M500 保存数值

(6) 风扇设置

```

# 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) 限位开关设置

```

#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           # "

```

如果要使用到限位开关，则“endstops_enable”这一项设置一定要为“true”。

Alpha, beta, gamma 对应三个轴。

Homing_direction 为回零的方向，回最小方向为 home_to_min，回最大方向为 home_to_max。

Alpha_min 为最小行程, alpha_max 为最大行程。其他轴同理。

(8) 自动调平设置

```

# 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

```

仅限于普通的接近开关设置。

调平过程，用上位机软件执行以下过程：

1) G32

#执行调平过程

2) G28

#归零

3) G0 Z5

#Z 轴上移 5mm

4) 手动调整 Z 轴高度，直到打印头和热床之间只有一张纸的厚度

5) M306 Z0

#设置当前高度为 0

6) G28

#再次归零

7) G0 Z1

#设置当前高度为 1mm，并测量打印头是否距离热床 1mm

8) M500

#将当前数据保存到 EEPROM 中；

#注意执行了 M500 之后，系统不会再从 config.txt 中读取参数，需要运行 M502 清除参数后，才会在下次启动时从 config.txt 中读取参数。

(9) LCD 显示屏设置

MKS LCD MINI12864/MKS LCD12864A (需移除 RPK2 排阻)

屏幕显示设置

```
# 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 sdc_d_pin 0.27!^ # sd detect signal (set to nc if no sdcard detect) (or any spar
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 屏幕设置

```
# 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
```

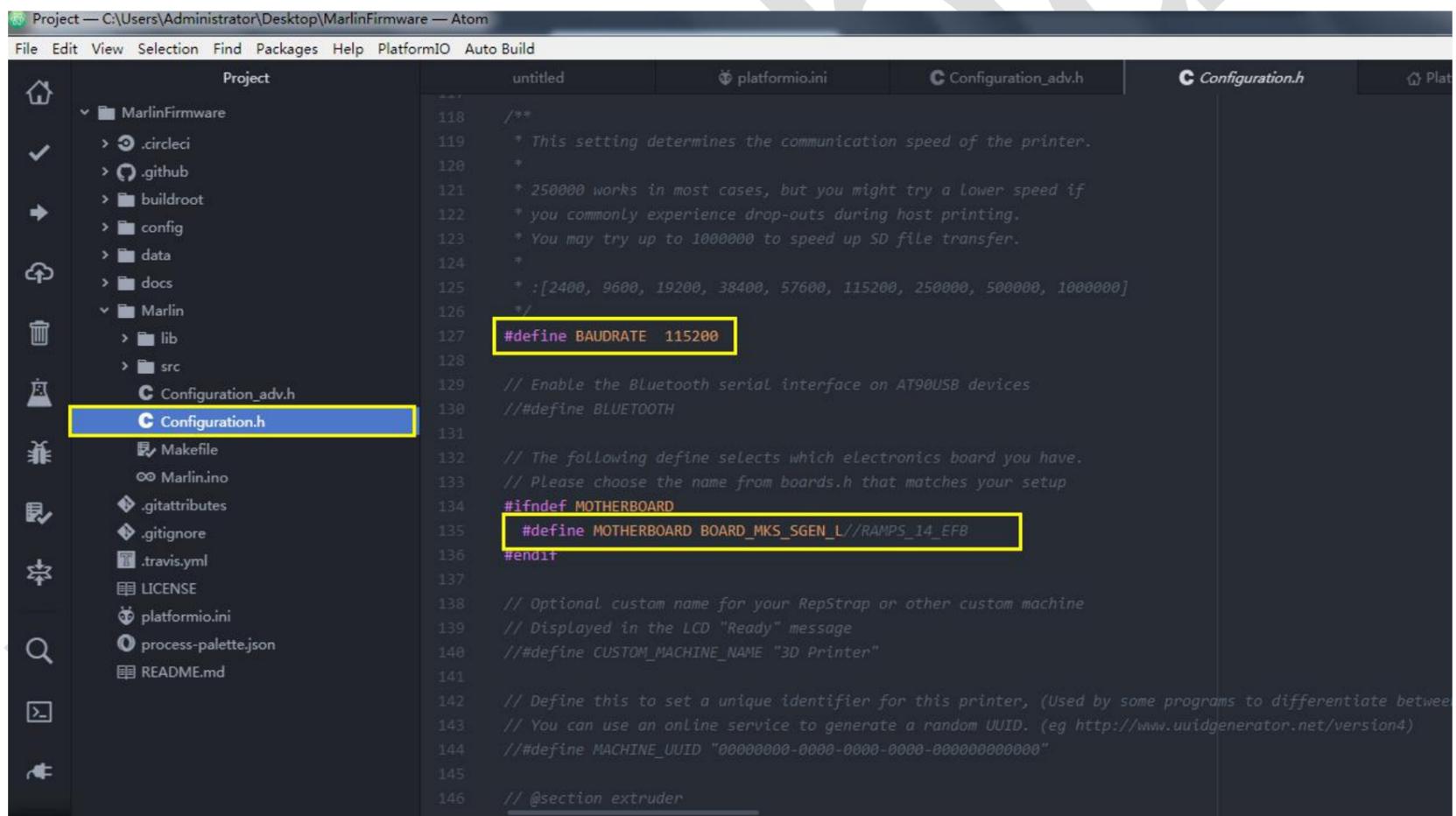
5.2 marlin2.0 参数配置

5.2.1 基本参数配置

Marlin2.0 参数设置可以直接通过 Atom 软件进行修改设置和编译

通过 atom 打开 marlin 固件，点击 File→Open Folder,打开 marlin2.0 固件的文件夹导入 marlin 固件。打开 configuration.h 文件，可进行基本的参数，设置内容和 1.0 版本的 marlin 的内容大致相同。

1. 主板波特率设置
2. 主板类型设置：主板类型设置为 BOARD_MKS_SGEN_L



3. 挤出头数量及热敏类型设置

The screenshot shows the Marlin Configuration.h file in a code editor. The left sidebar displays the project structure, with Configuration.h selected. The main editor area shows the following code snippet:

```

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 // {0, 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 */

```

The screenshot shows the Marlin Configuration.h file in a code editor. The left sidebar displays the project structure, with Configuration.h selected. The main editor area shows the following code snippet:

```

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 V
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

```

根据使用温度传感器类型进行设置

其中#define TEMP_SENSOR_BED 1（此项为热床设置项，设置位 0 则为不使能热床，设为其他数字则为对应的传感器）

4. 温度设置

最高温度与最低温度设置（如果检测到实际温度不在这个范围之内，主板将会进行报错）

```

409 #define TEMP_HYSTERESIS 5 // (°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 设置（一般不用进行调节，默认即可）

调节方法和 smoothieware 相同。使用 M303 指令，例如：

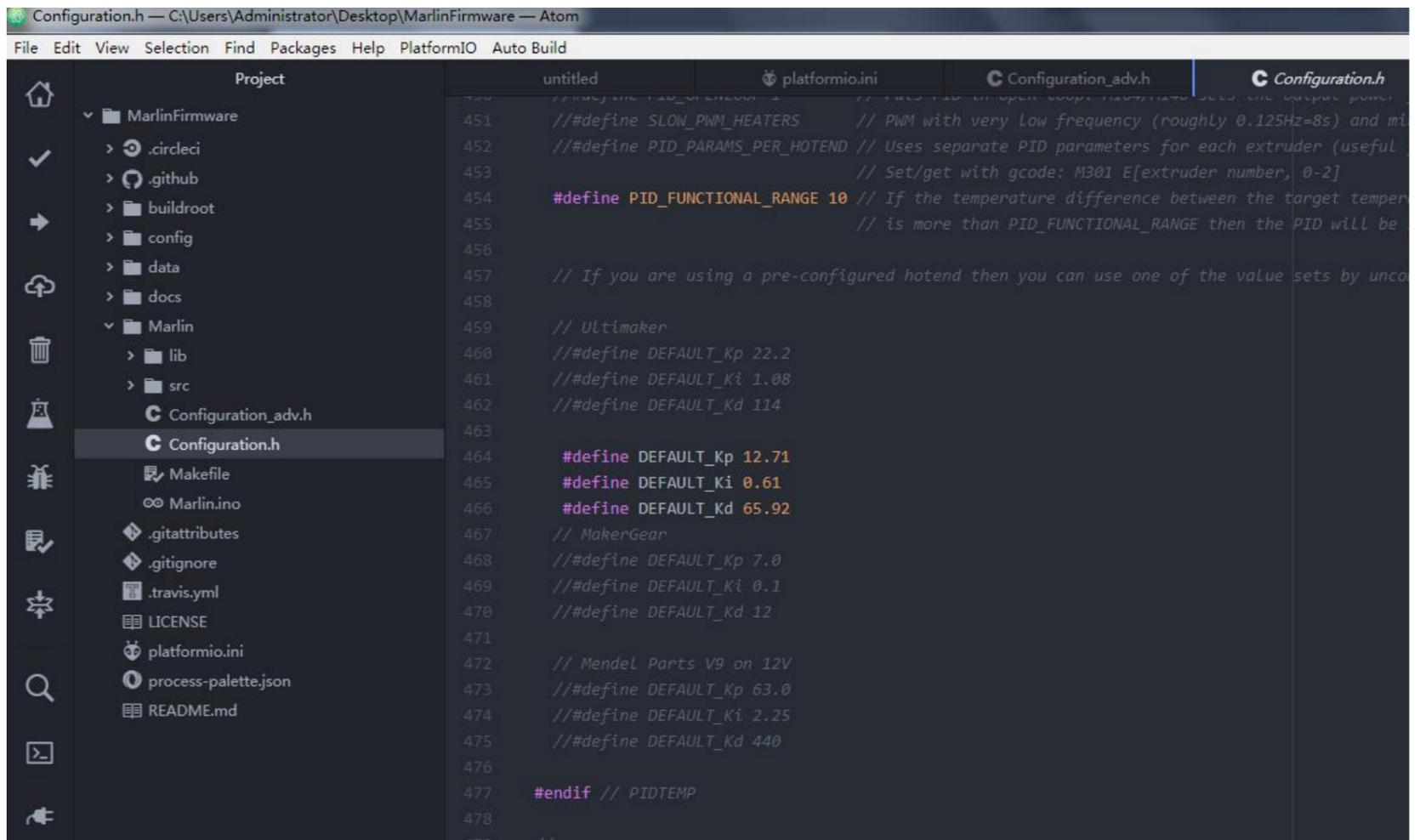
```
M303 E0 S190
```

针对打印头 E0 经常加热到 190 度的情况，自动运行 PID 运算。运算完成之后，将返回的数值填写进固件之中即可以

```

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

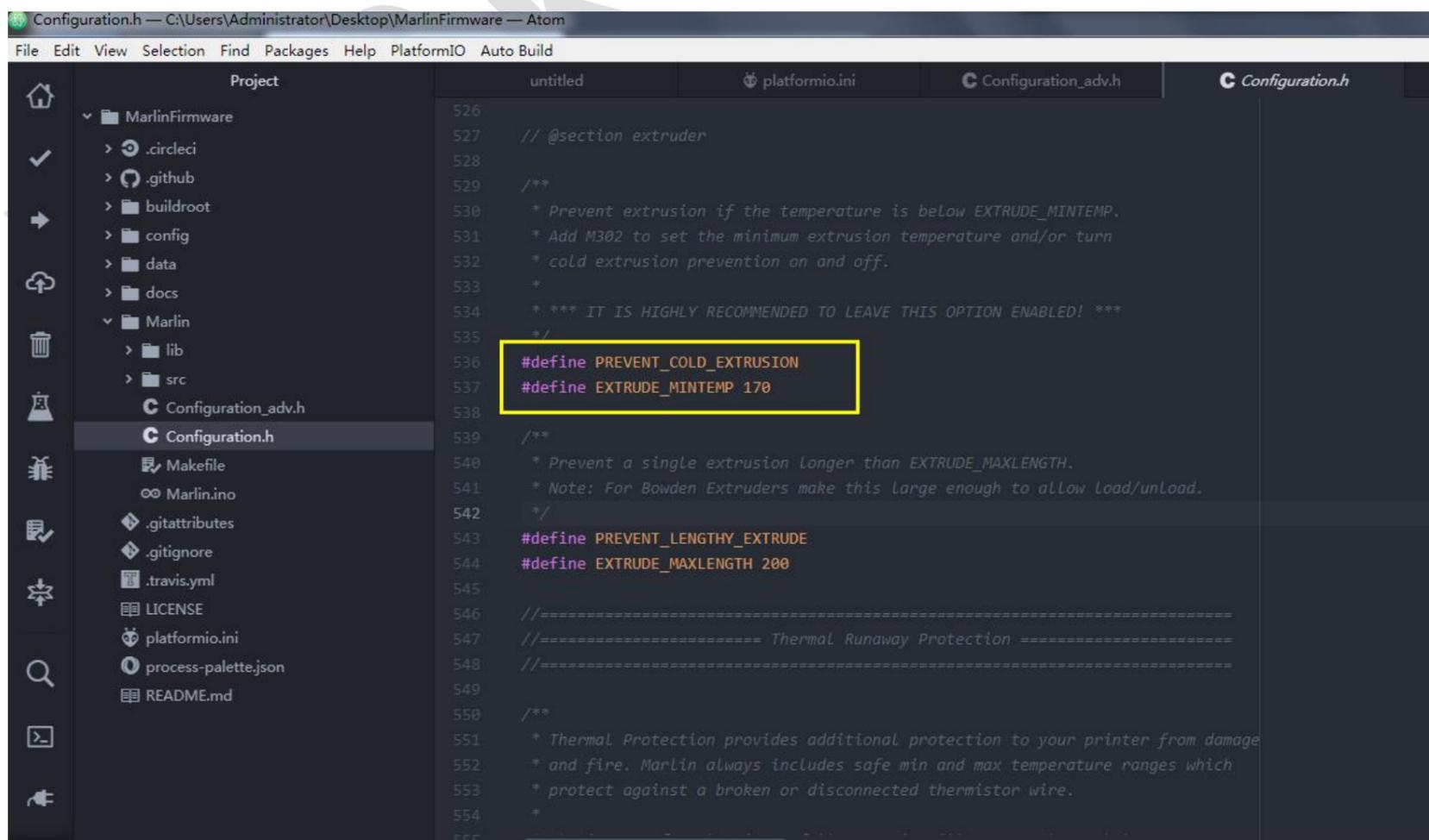
```



```
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
```

冷挤出保护设置（PREVENT_COLD_EXTRUSION）

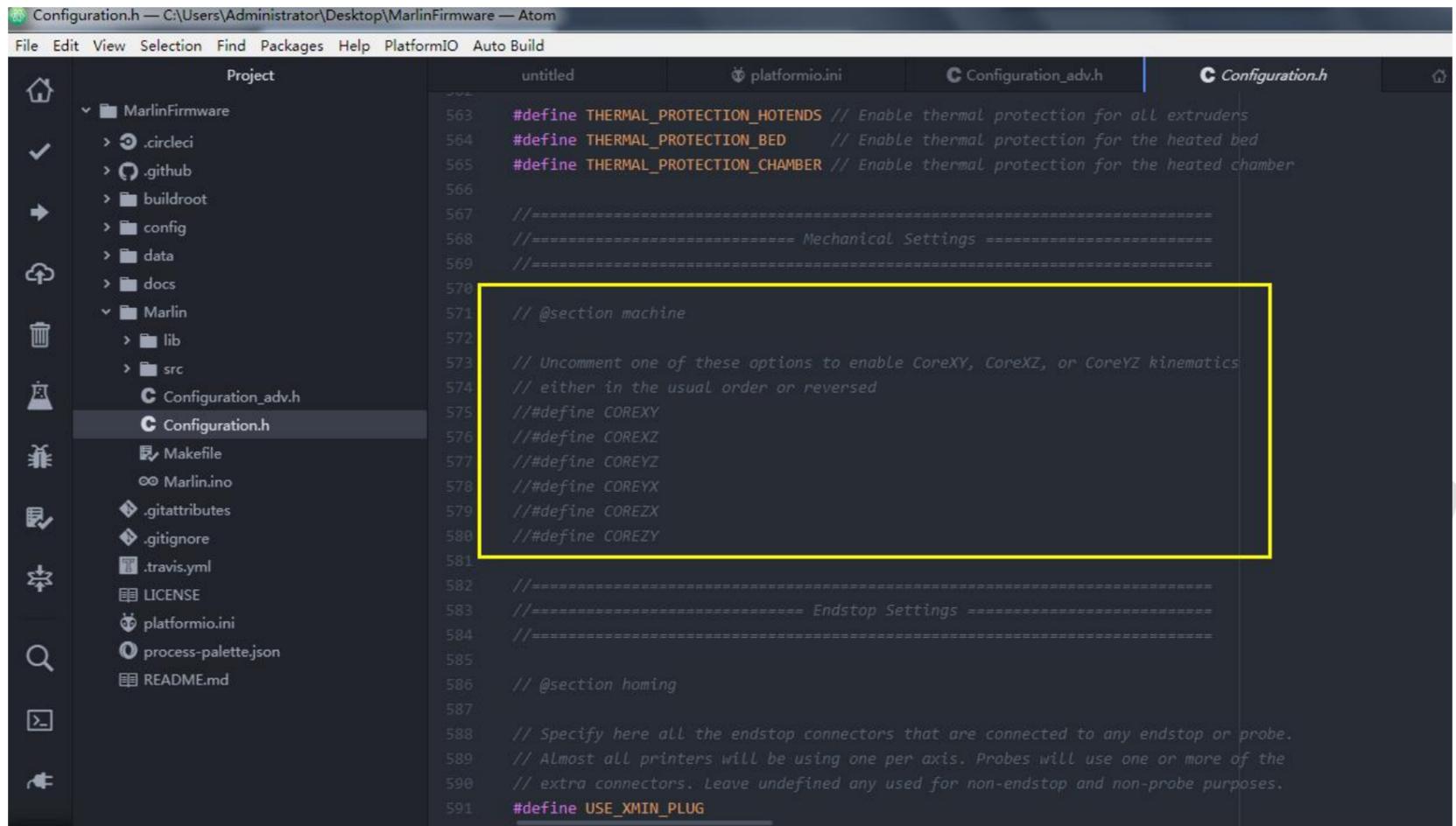
默认为 170 度，挤出电机需要加热头加到 170 度或者以上才能使电机进行挤出。如果机器挤出不需要加热，则可以将此温度调低即可。



```
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 //===== Thermal Runaway Protection =====
547 //=====
548
549 /**
550  * Thermal Protection provides additional protection to your printer from damage
551  * and fire. Marlin always includes safe min and max temperature ranges which
552  * protect against a broken or disconnected thermistor wire.
553  *
554  *
```

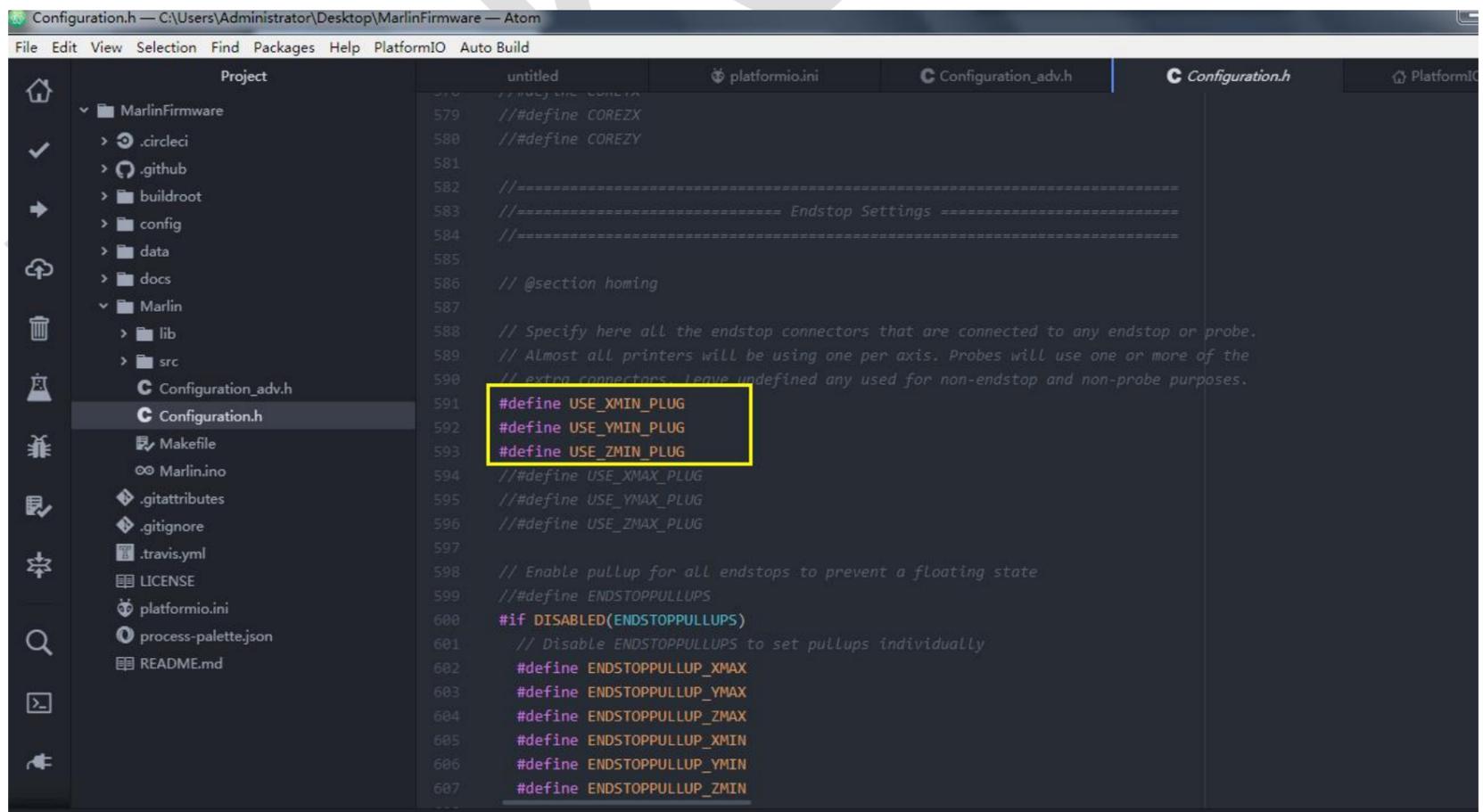
5. 机器设置

机器结构设置，默认为 xyz 结构，如果是 corexy 或者是其他的结构则需要使能相应的类型



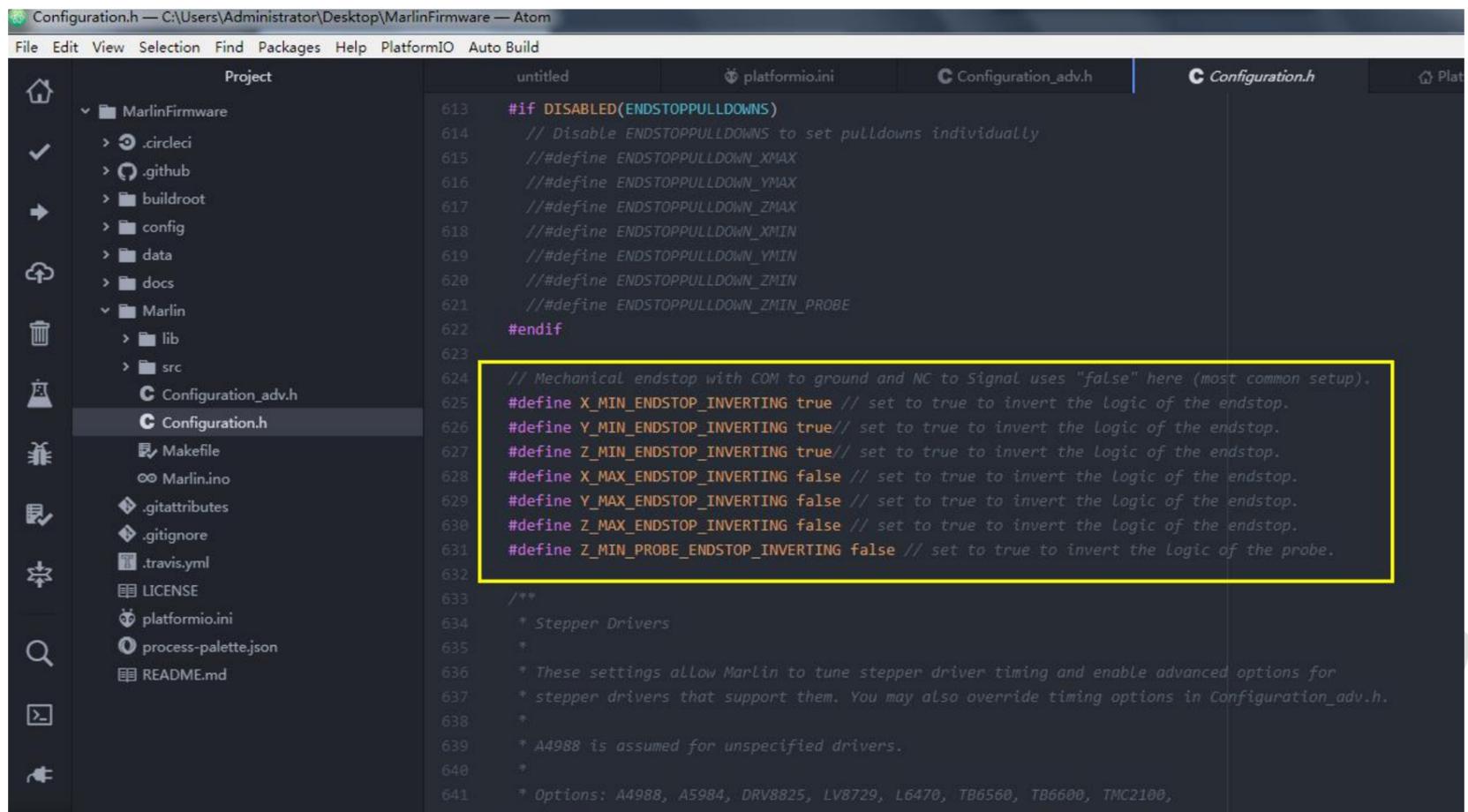
限位开关设置

使能限位开关



设置开关类型（常开/常闭）

注：如果设置类型相反，可能或导致回零不正常

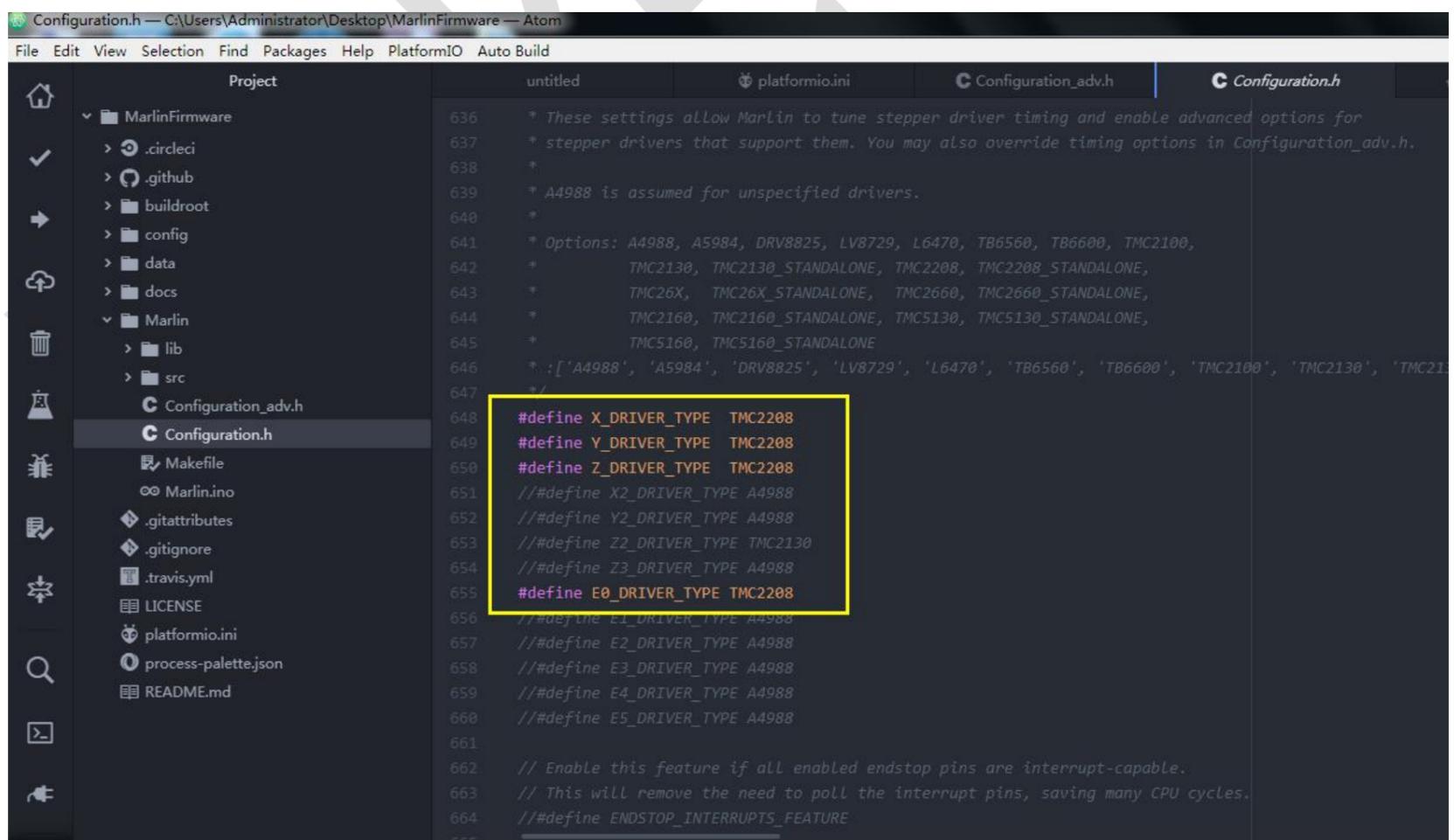


```

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,
  
```

驱动类型设置

设置使用的驱动类型，才能配置相应的特殊模式（tmc2208 的 uart 模式，tmc2130 的 SPI 模式）如果没有对几个驱动类型项进行使能的话，则所用驱动均为普通模式。



```

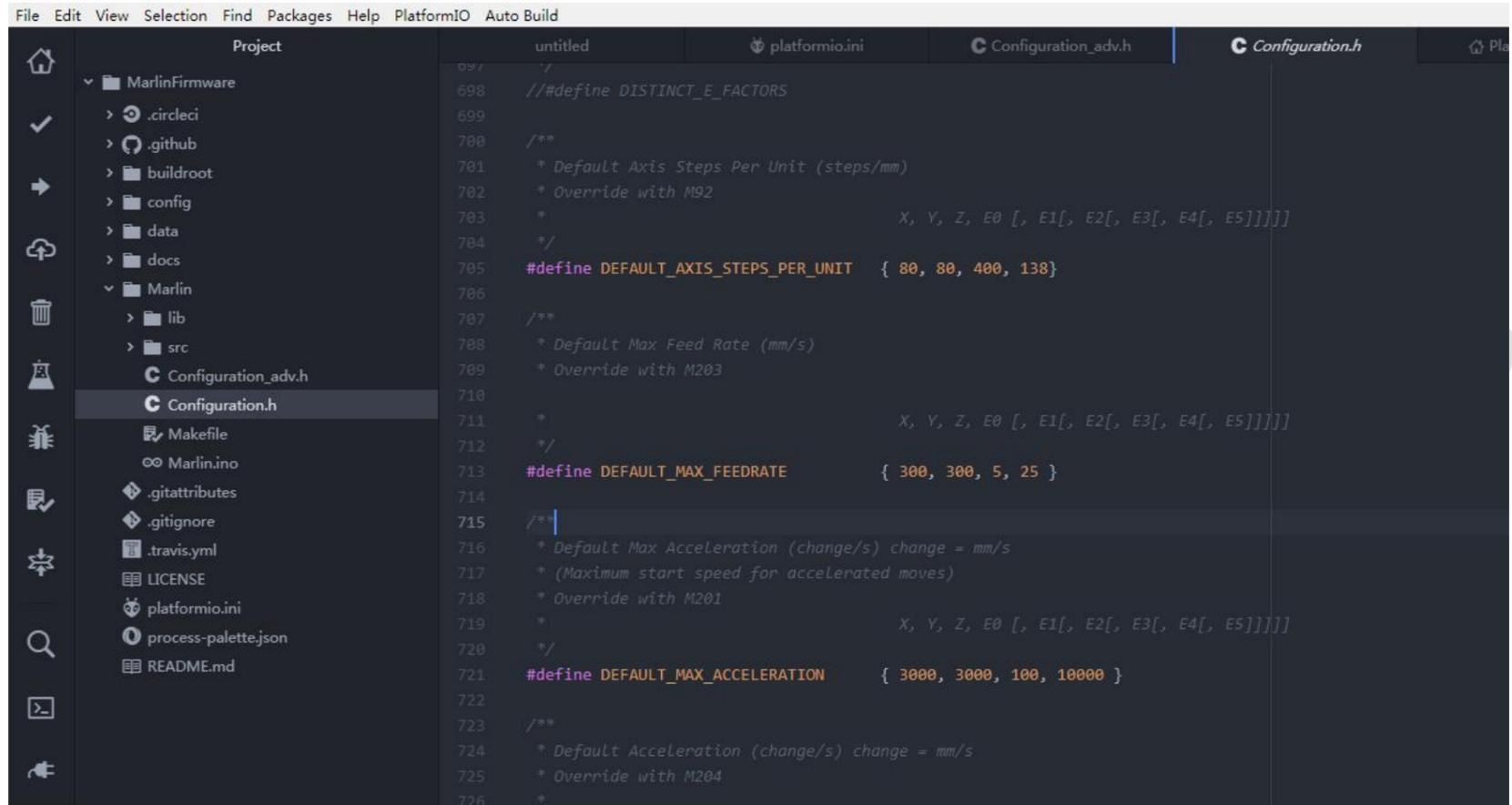
636 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
637 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
638 *
639 * A4988 is assumed for unspecified drivers.
640 *
641 * Options: A4988, A5984, DRV8825, LV8729, 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
  
```

电机脉冲设置（#define DEFAULT_AXIS_STEPS_PER_UNIT）

可根据机械情况，通过公式进行计算

同步轮电机脉冲数/mm 的公式为： $(360 \div \text{步距}) \times \text{细分} \div (\text{直径} \times 3.14)$ ；

丝杆电机脉冲数/mm 的公式为： $(360 \div \text{步距}) \times \text{细分} \div \text{导程}$ 。



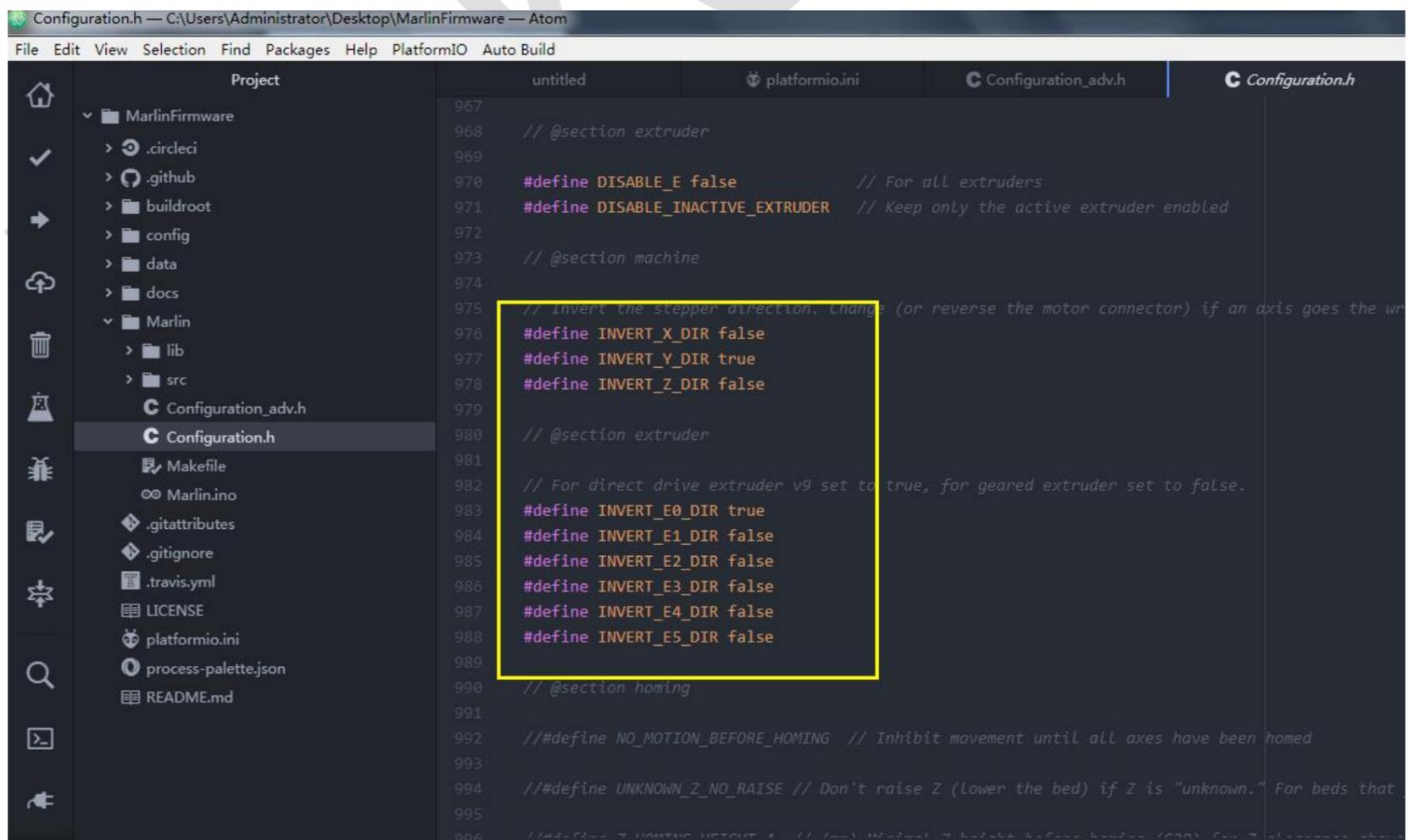
```

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  */

```

电机方向设置

可以通过调整设置 true 和 false 进行调整电机方向



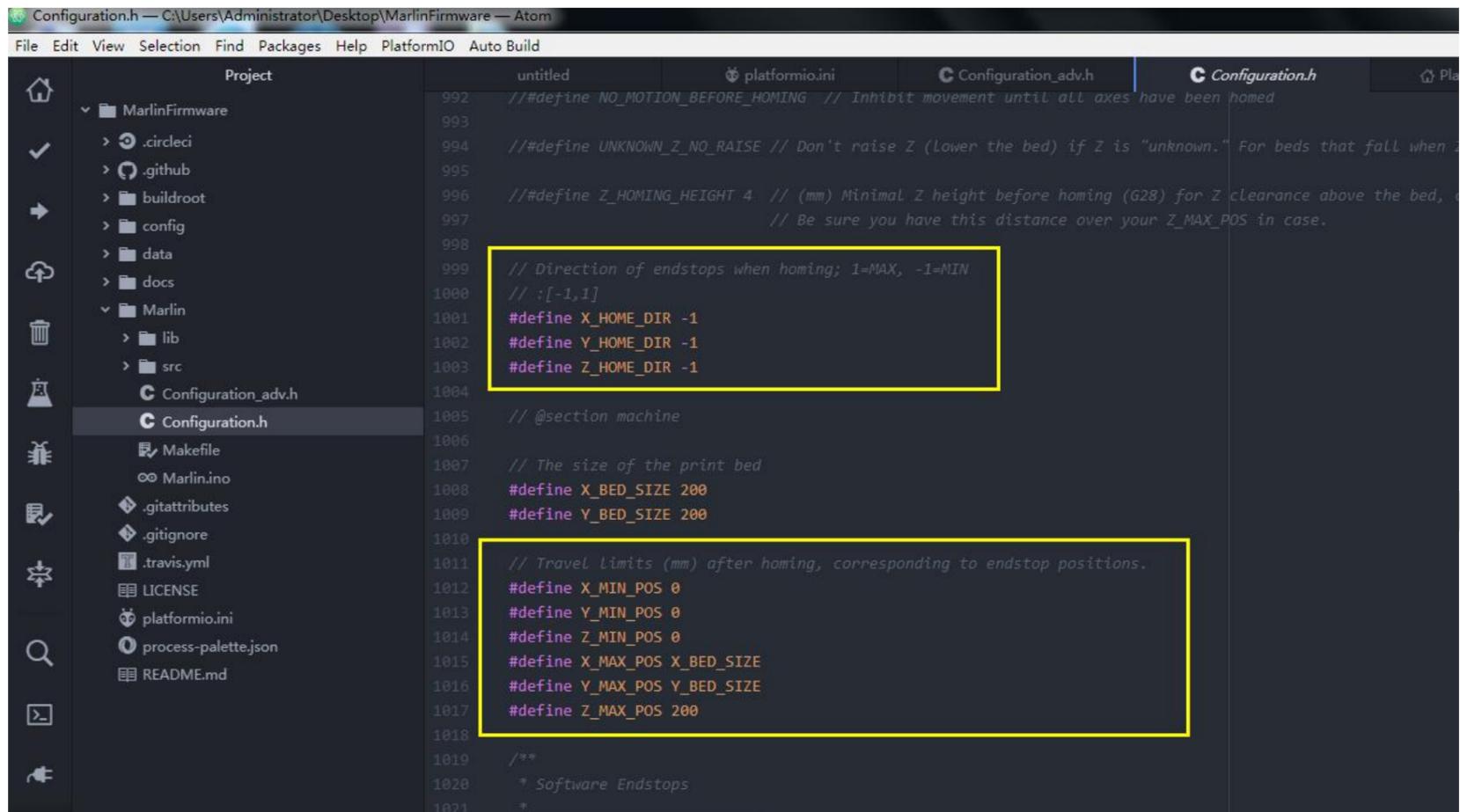
```

967 // @section extruder
968
969 #define DISABLE_E false // For all extruders
970 #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
971
972 // @section machine
973
974 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
975
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 // don't raise Z on homing. (e.g. Z-axis has a spring to force down when not powered) See also: https://www.printer-

```

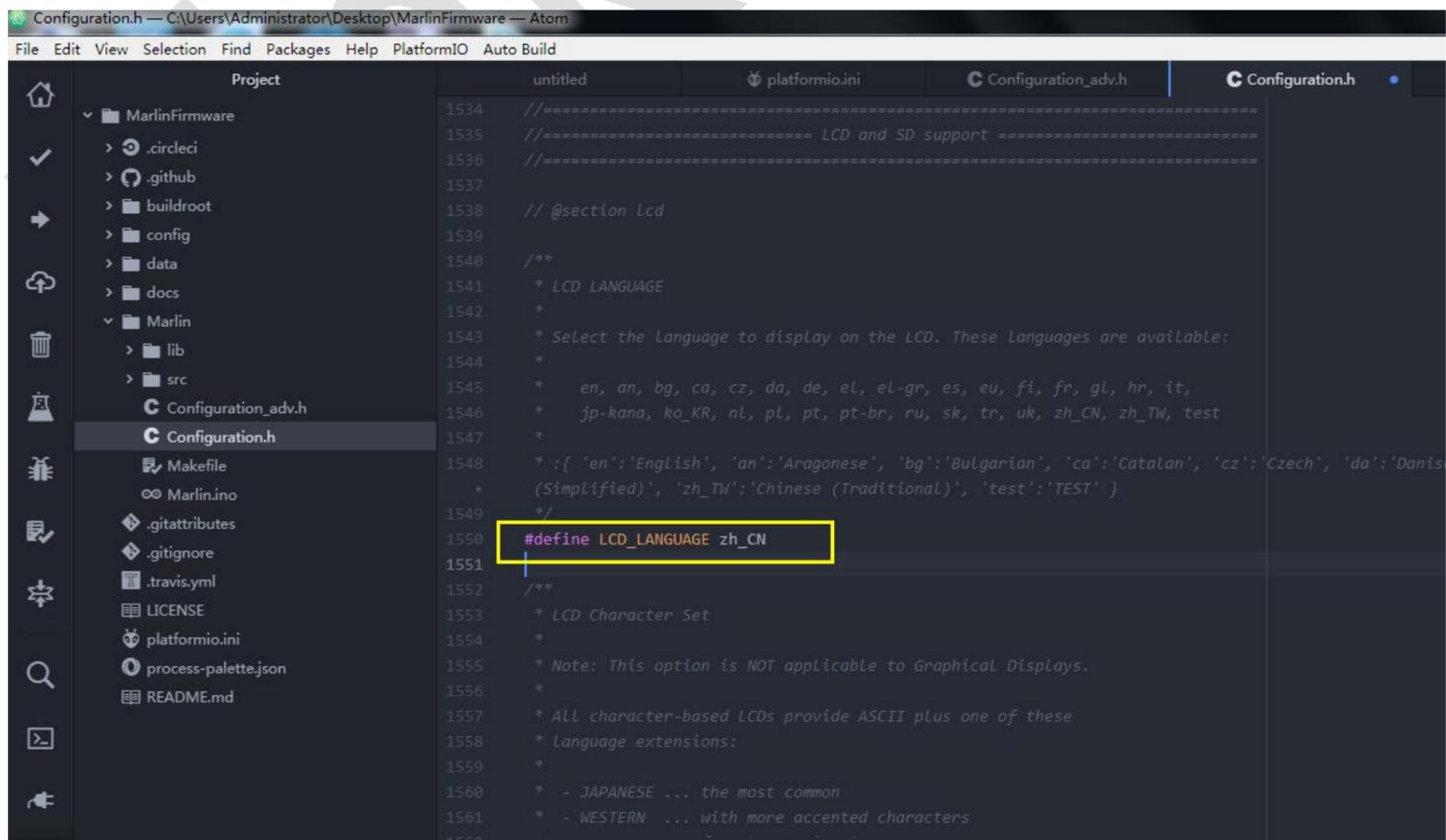
回零方向设置：-1 为回最小方向回零，1 为最大方向回零。

行程设置：设置 xyz 轴最大的运行距离。



6. LCD 屏幕设置

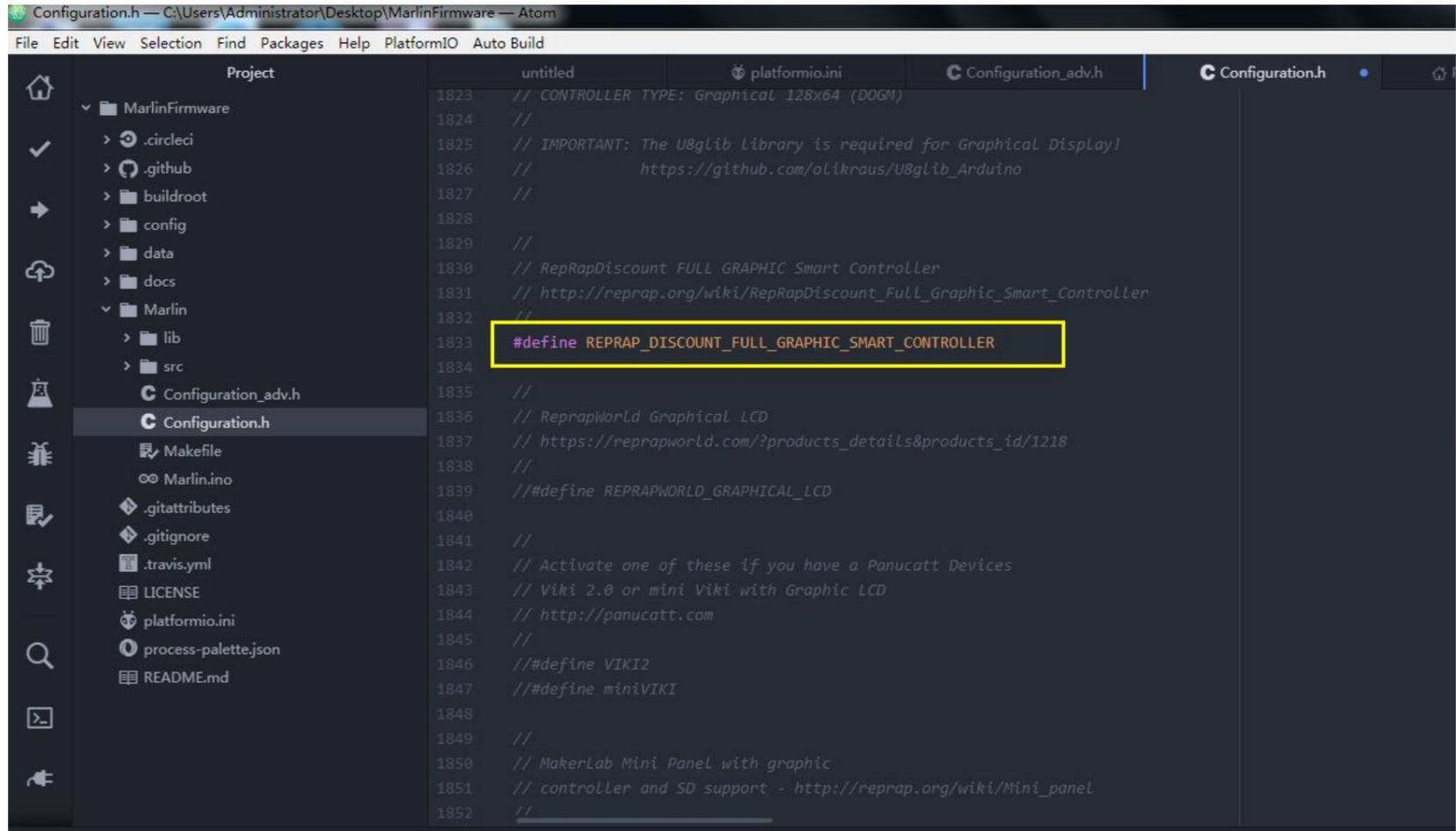
语言设置：zh_CN 为简体中文，en 为英文



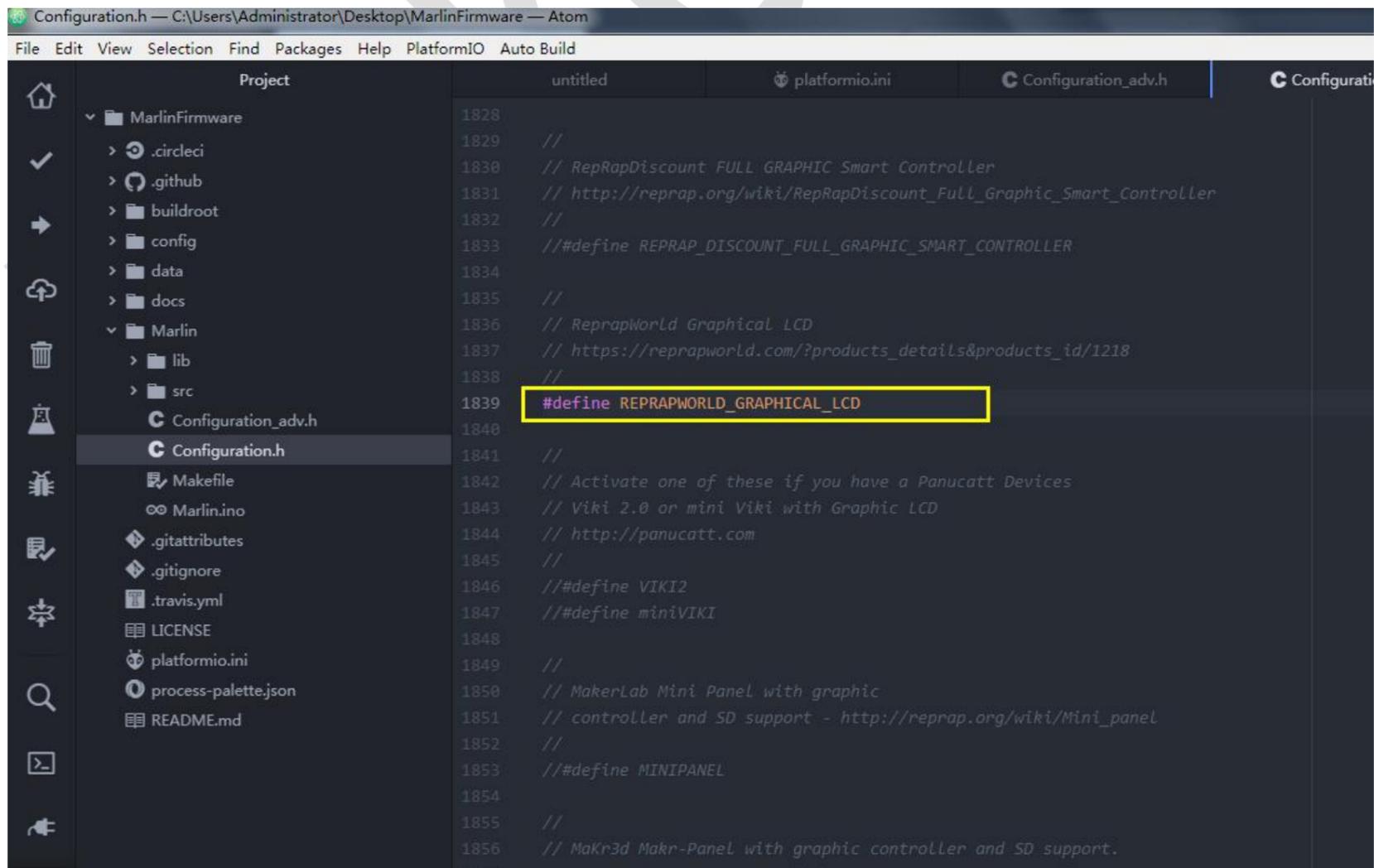
屏幕类型使能

固件中只能使能一种 lcd 屏幕，如果使能多种，容易造成错误

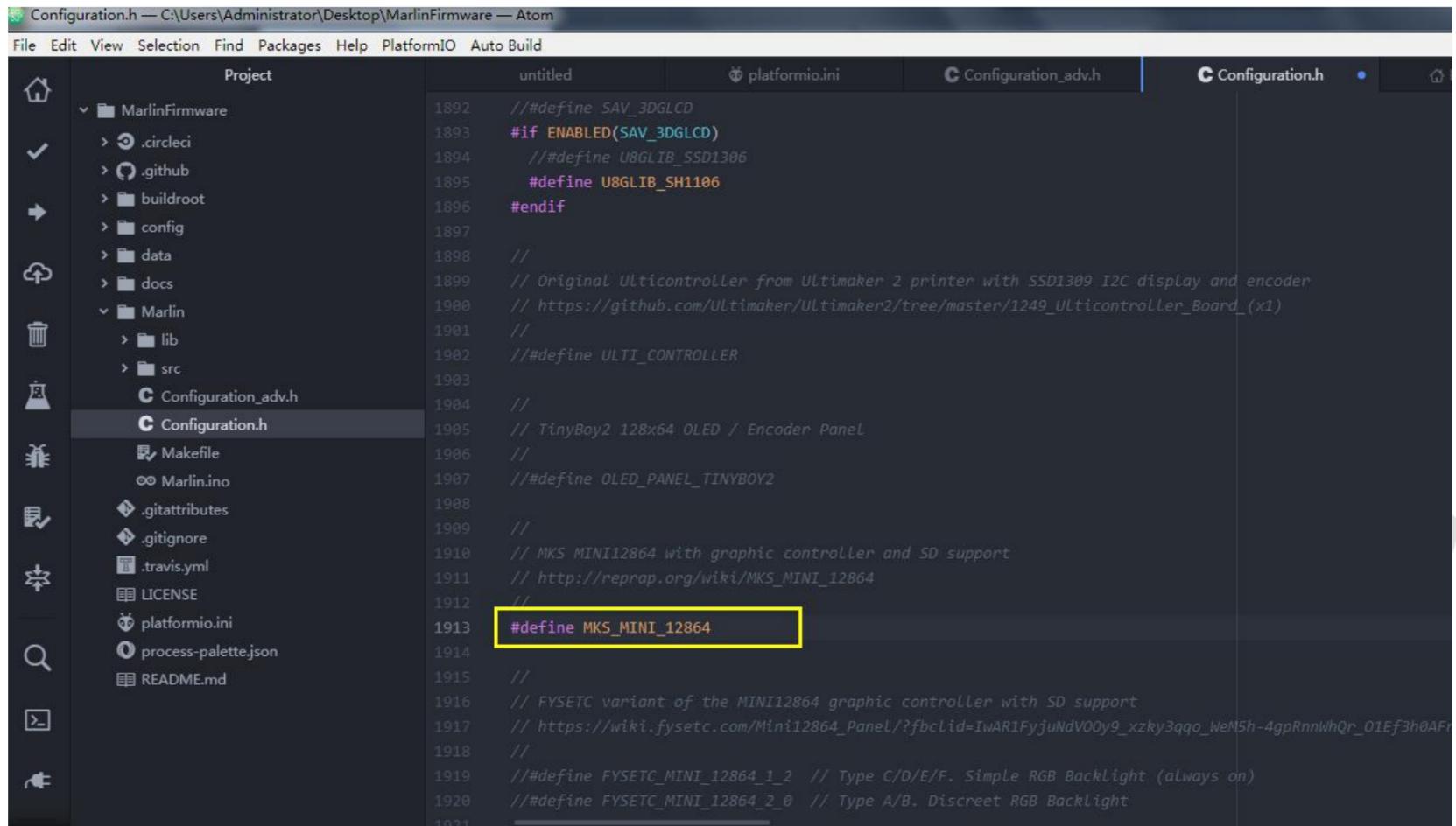
LCD12864 使能



LCD2004 使能



Mini12864 和 LCD12864A 使能



```
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_xzky3qqa_wel5h-4gpRnnWhQr_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
```

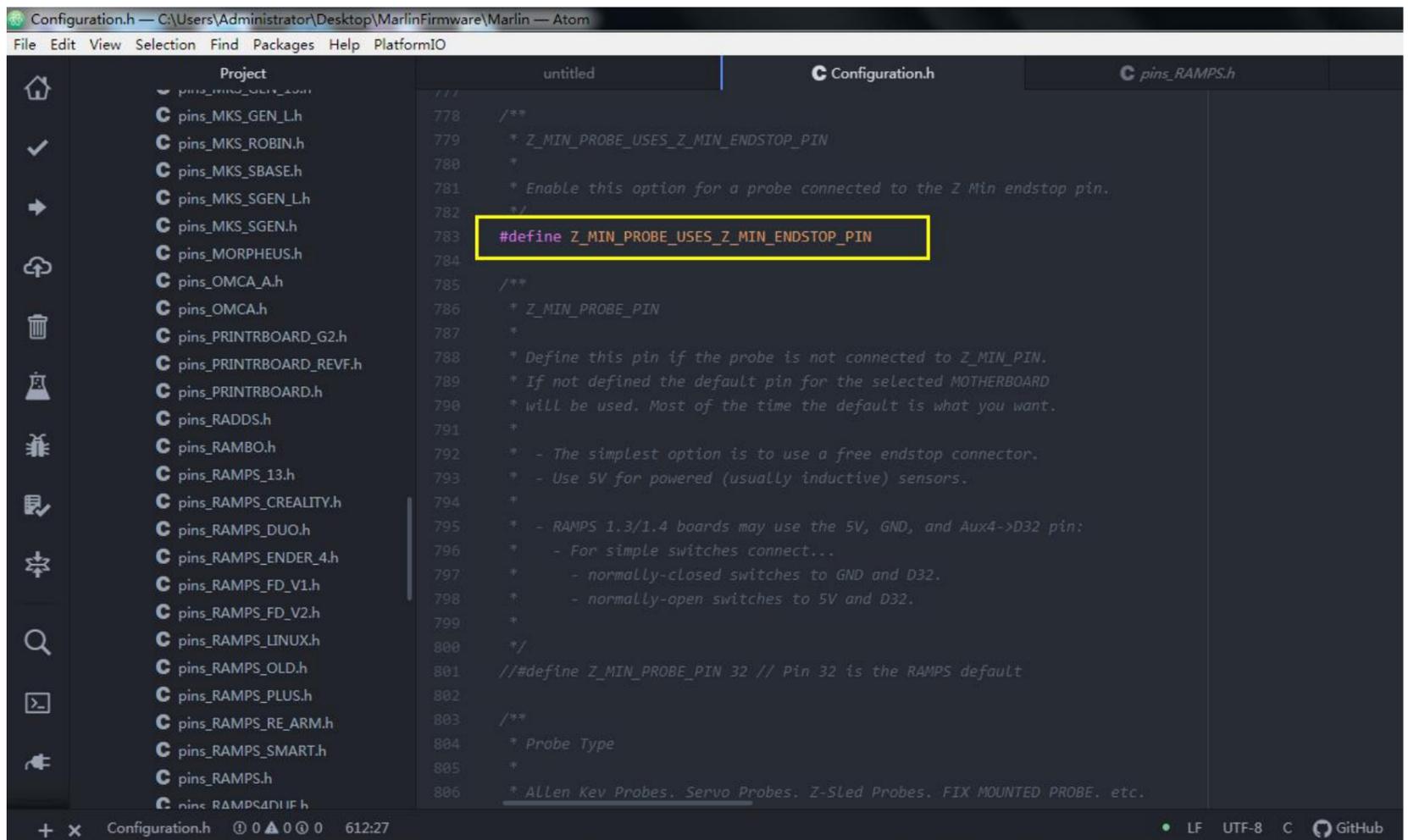
Makerbase

5.2.2 自动调平设置（z-probe 和 3d touch）

在马林 2.0 固件上的配置主要参考《3Dtouch 使用说明书》

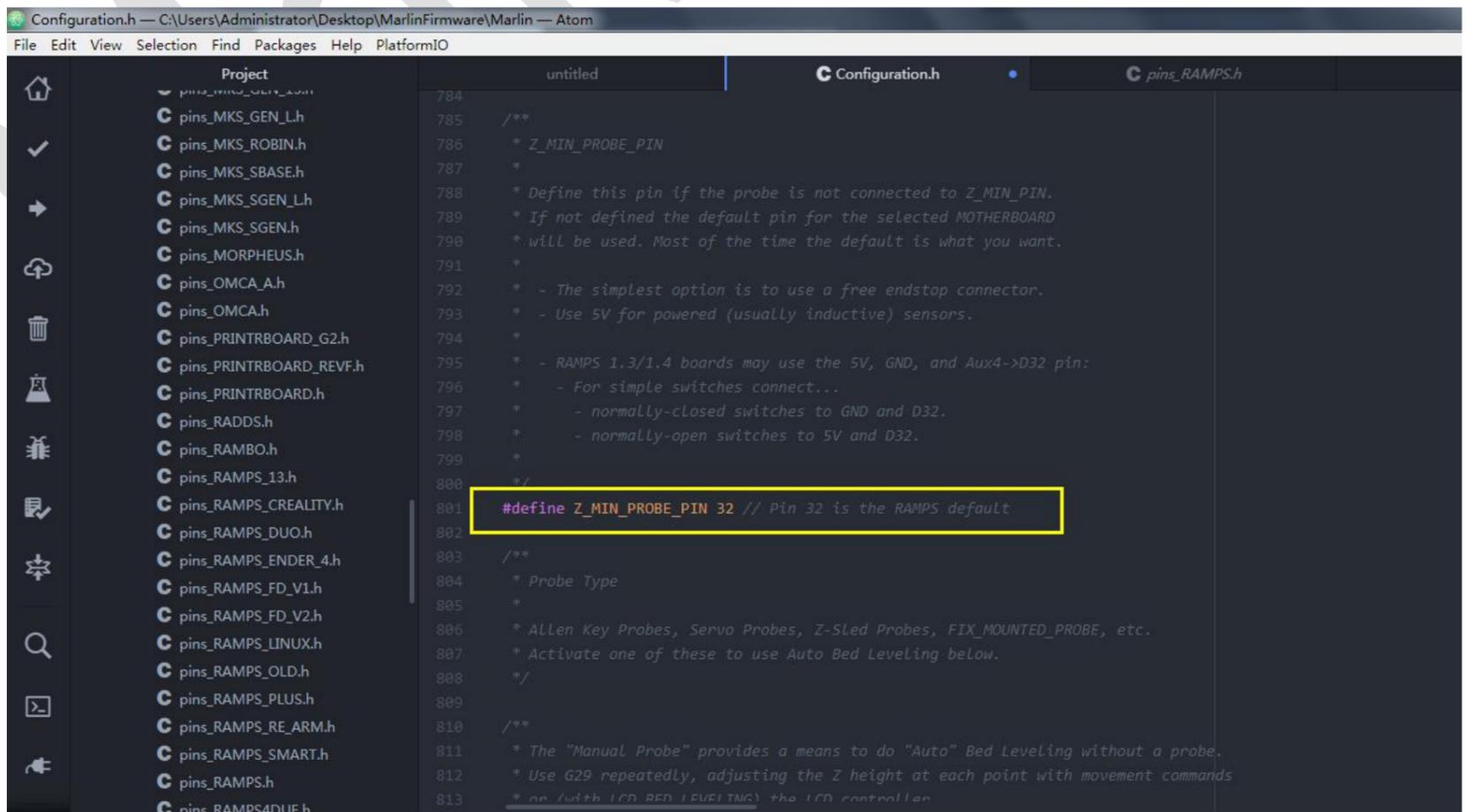
如果使用使用 Z-限位作为调平开关的触发限位的，则使能这一项

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

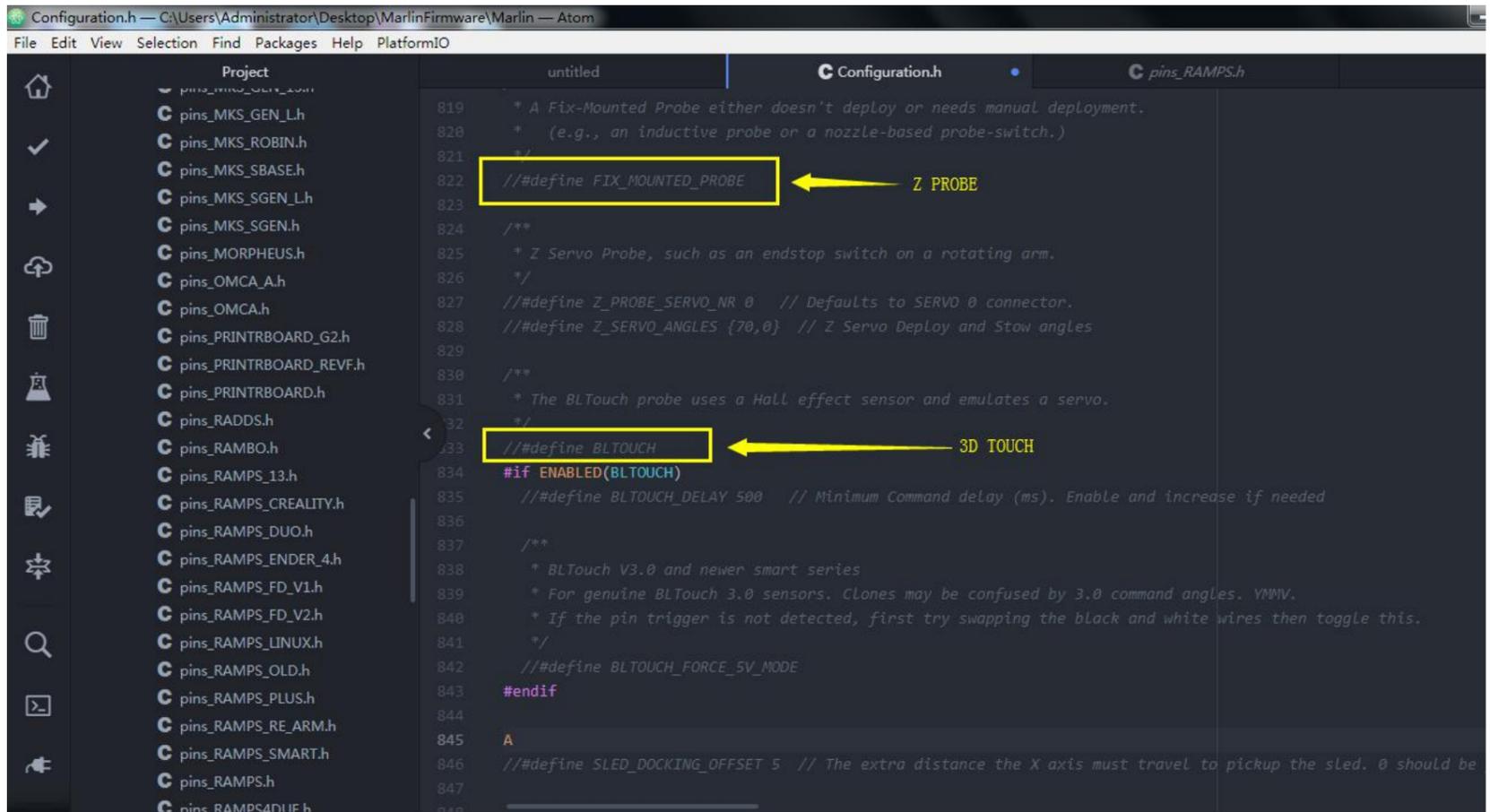


如果想使能在其他限位的话，则可以使能这一项，并修改对应限位的 pin 脚。

```
#define Z_MIN_PROBE_PIN 32
```

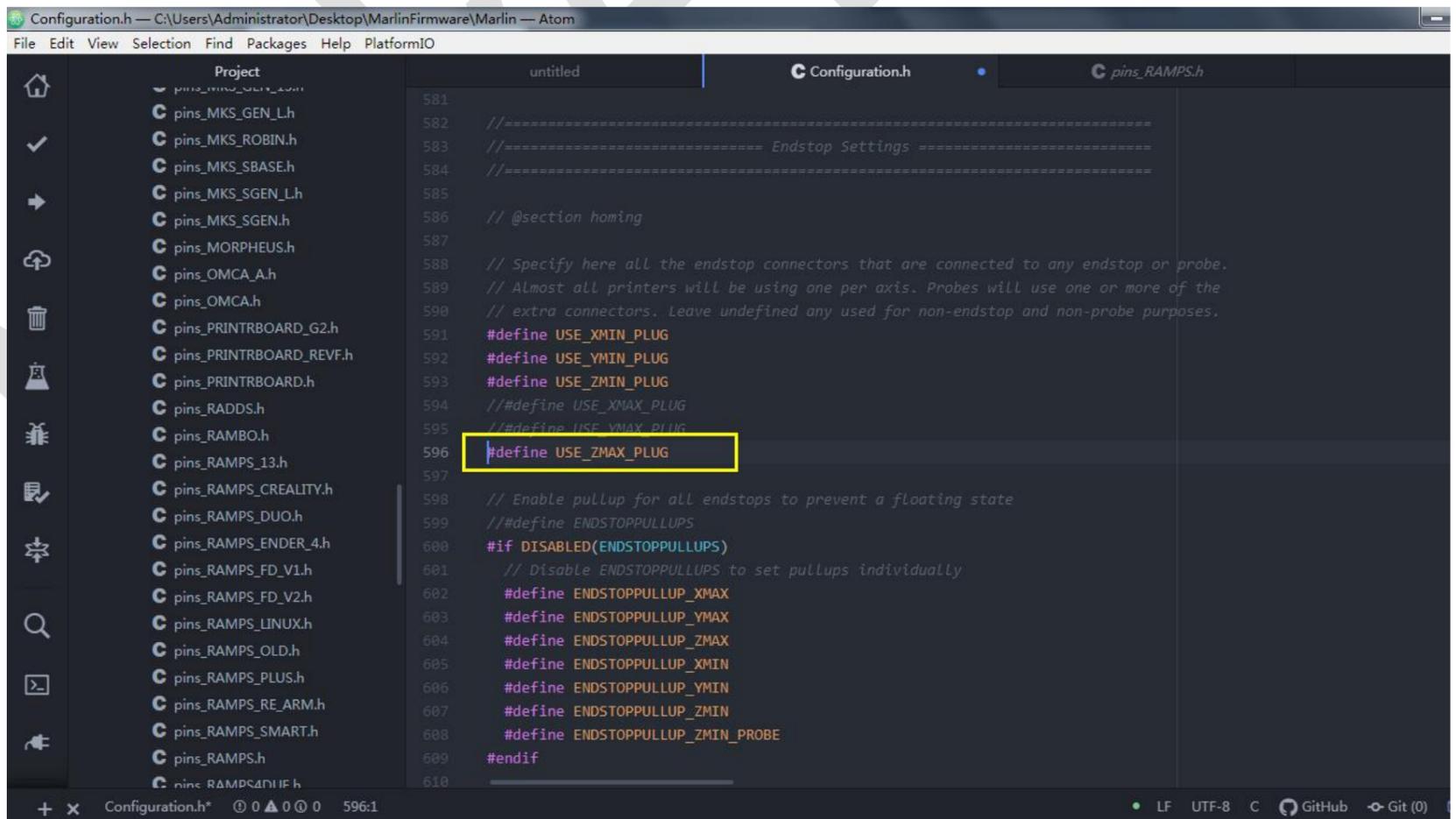


如果使用 Z-probe 则使能上面这一项，如果使用 3dtouch 则使用下面那一项。不能同时使能多项，如果使能一项需要将另外屏蔽掉



如果调平开关有使用到相关限位的话，必须对其进行使能。

开关的常开常闭类型也要进行设置，一般默认为 false。



```

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 /**
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  * Optima: A4988 A5084 DRV8825 LV1770 L6470 TRAF60 TRAF60 TMC2102

```

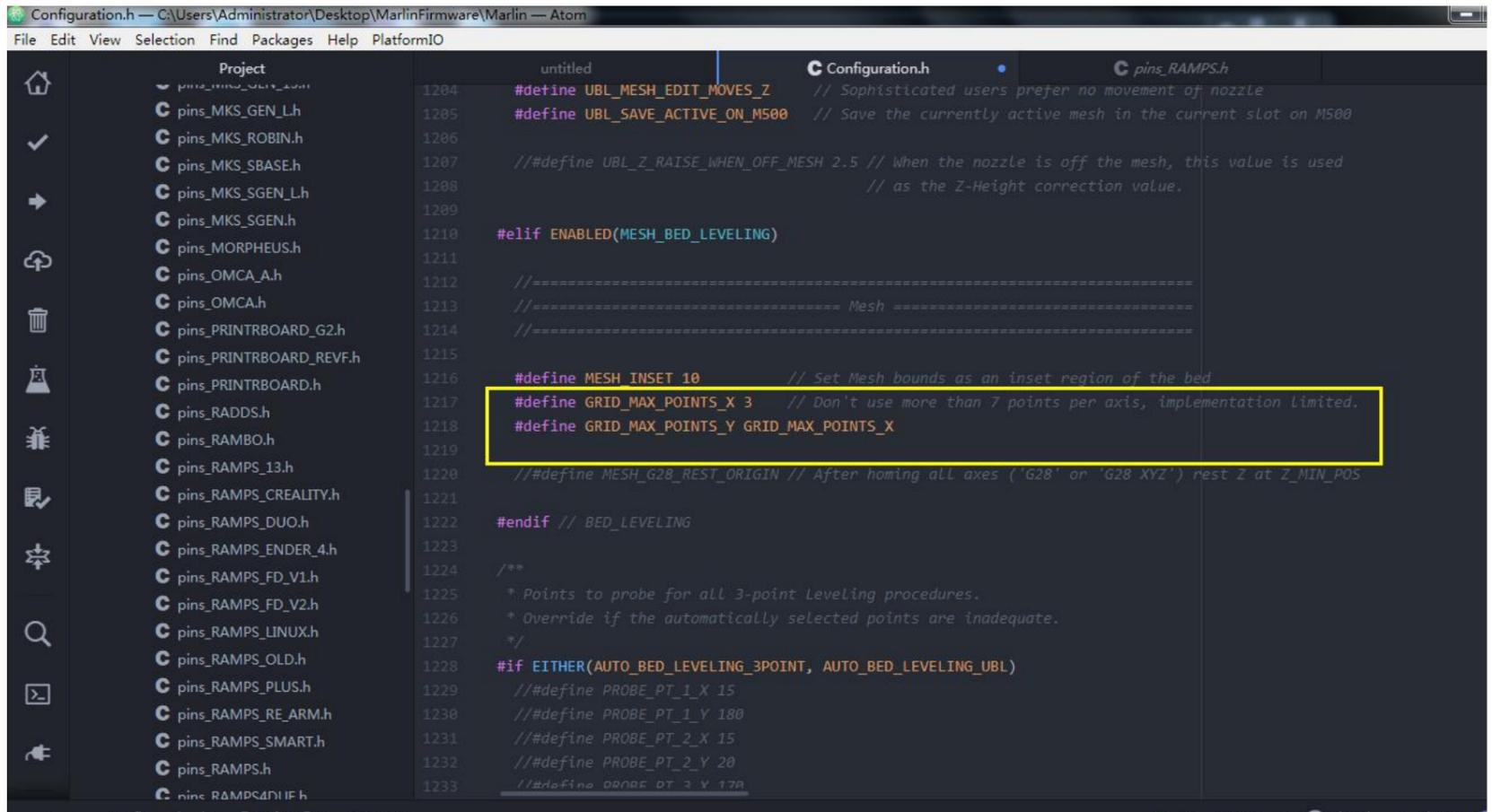
选择调平模式，使能#define AUTO_BED_LEVELING_BILINEAR

```

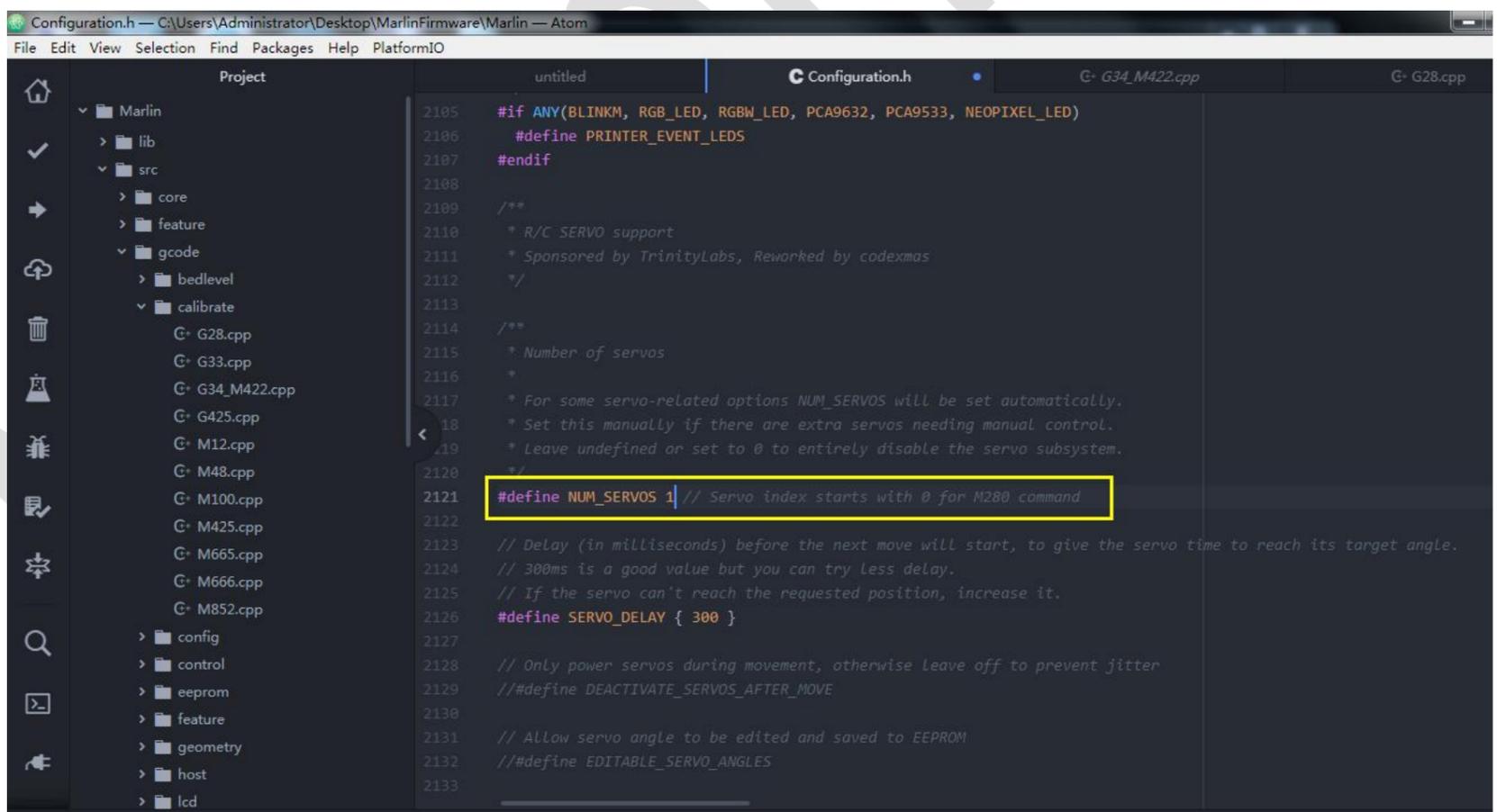
1104 * of other systems. UBL also includes integrated Mesh Generation, Mesh
1105 * Validation and Mesh Editing systems.
1106 *
1107 * - MESH_BED_LEVELING
1108 * Probe a grid manually
1109 * The result is a mesh, suitable for large or uneven beds. (See BILINEAR.)
1110 * For machines without a probe, Mesh Bed Leveling provides a method to perform
1111 * leveling in steps so you can manually adjust the Z height at each grid-point.
1112 * With an LCD controller the process is guided step-by-step.
1113 */
1114 // #define AUTO_BED_LEVELING_3POINT
1115 // #define AUTO_BED_LEVELING_LINEAR
1116 #define AUTO_BED_LEVELING_BILINEAR
1117 // #define AUTO_BED_LEVELING_UBL
1118 // #define MESH_BED_LEVELING
1119
1120 /**
1121  * Normally G28 leaves leveling disabled on completion. Enable
1122  * this option to have G28 restore the prior leveling state.
1123  */
1124 // #define RESTORE_LEVELING_AFTER_G28
1125
1126 /**
1127  * Enable detailed logging of G28, G29, M48, etc.
1128  * Turn on with the command 'M111 S32'.
1129  * NOTE: Requires a lot of PROGMEM!
1130  */
1131 // #define DEBUG_LEVELING_FEATURE
1132

```

调平点数设置



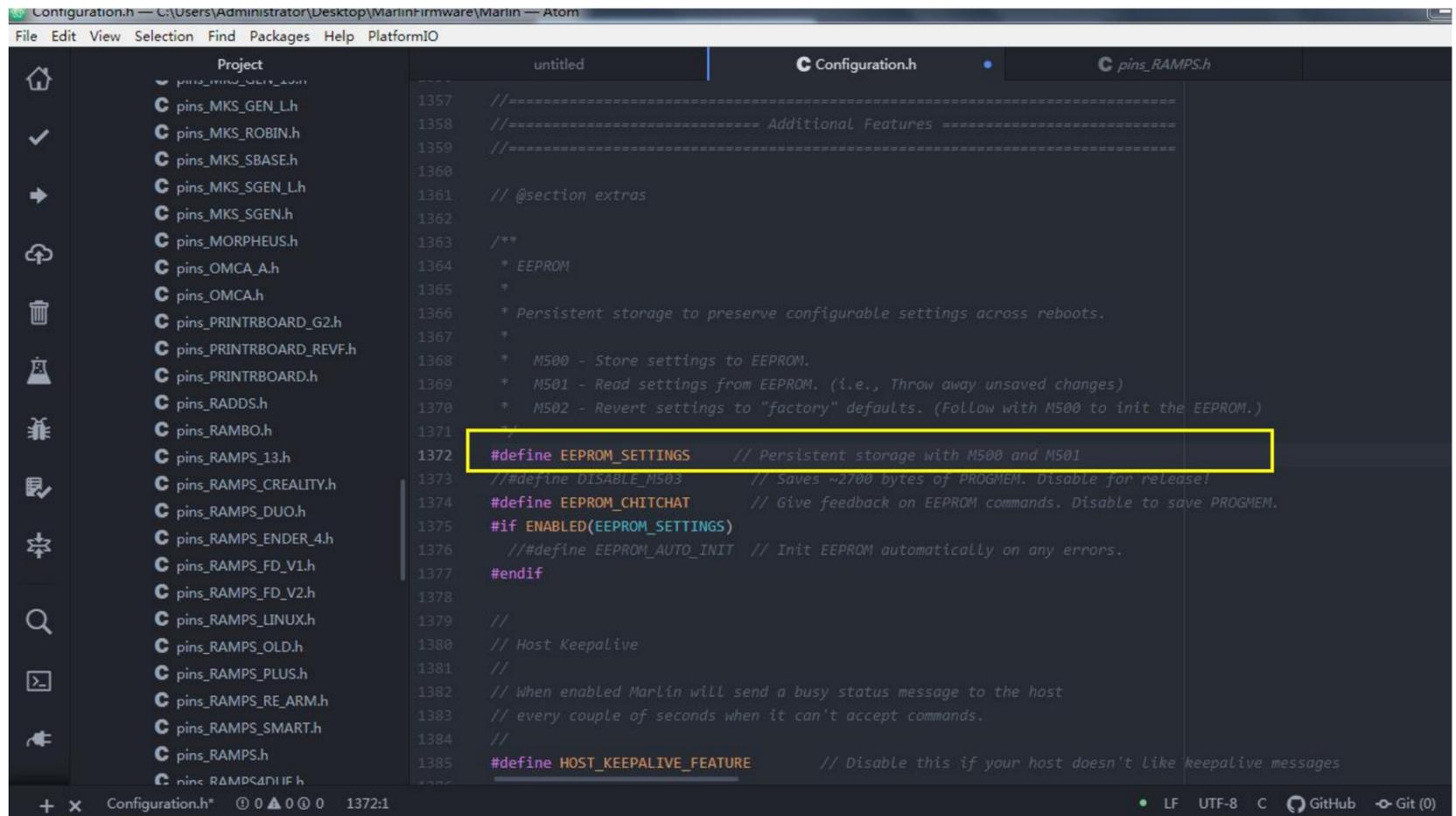
开启舵机数量并数量改为 1



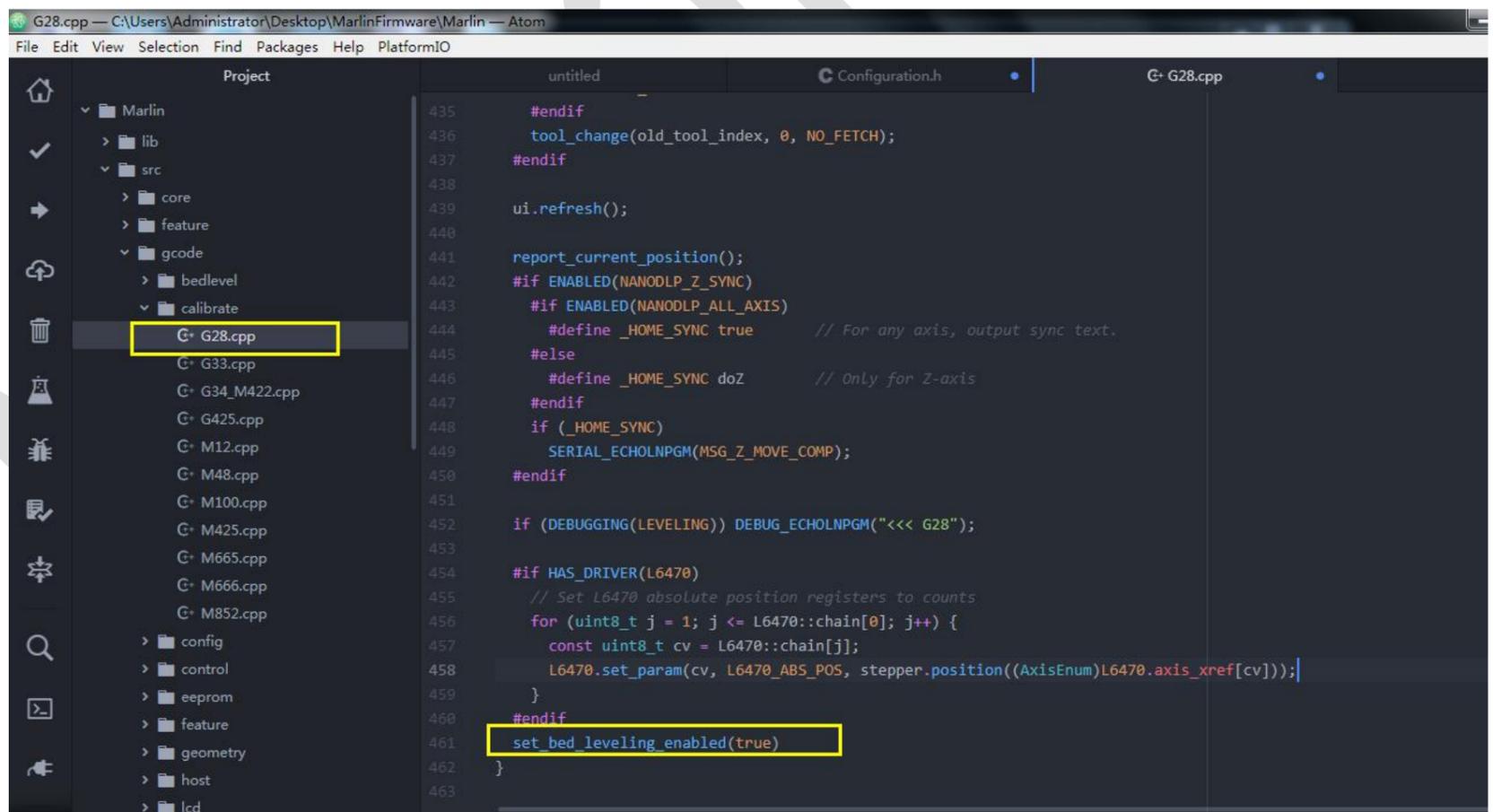
调平数据保存设置

需先使能`#define EEPROM_SETTINGS`

到 `gcode-calibrate-G28.cpp` 文件中添加 `set_bed_leveling_enabled(true)`, 如图所示。



```
1357 //=====
1358 //===== Additional Features =====
1359 //=====
1360
1361 // @section extras
1362
1363 /**
1364  * EEPROM
1365  *
1366  * Persistent storage to preserve configurable settings across reboots.
1367  *
1368  * M500 - Store settings to EEPROM.
1369  * M501 - Read settings from EEPROM. (i.e., Throw away unsaved changes)
1370  * M502 - Revert settings to "factory" defaults. (Follow with M500 to init the EEPROM.)
1371  */
1372 #define EEPROM_SETTINGS // Persistent storage with M500 and M501
1373 // #define DISABLE_M503 // Saves ~2700 bytes of PROGMEM. Disable for release!
1374 #define EEPROM_CHITCHAT // Give feedback on EEPROM commands. Disable to save PROGMEM.
1375 #if ENABLED(EEPROM_SETTINGS)
1376 // #define EEPROM_AUTO_INIT // Init EEPROM automatically on any errors.
1377 #endif
1378
1379 //
1380 // Host Keepalive
1381 //
1382 // When enabled Marlin will send a busy status message to the host
1383 // every couple of seconds when it can't accept commands.
1384 //
1385 #define HOST_KEEPALIVE_FEATURE // Disable this if your host doesn't like keepalive messages
```



```
435 #endif
436 tool_change(old_tool_index, 0, NO_FETCH);
437 #endif
438
439 ui.refresh();
440
441 report_current_position();
442 #if ENABLED(NANODLP_Z_SYNC)
443 #if ENABLED(NANODLP_ALL_AXIS)
444 #define _HOME_SYNC true // For any axis, output sync text.
445 #else
446 #define _HOME_SYNC doZ // Only for Z-axis
447 #endif
448 if (_HOME_SYNC)
449 SERIAL_ECHOLNPGM(MSG_Z_MOVE_COMP);
450 #endif
451
452 if (DEBUGGING(LEVELING)) DEBUG_ECHOLNPGM("<<<< G28");
453
454 #if HAS_DRIVER(L6470)
455 // Set L6470 absolute position registers to counts
456 for (uint8_t j = 1; j <= L6470::chain[0]; j++) {
457 const uint8_t cv = L6470::chain[j];
458 L6470.set_param(cv, L6470_ABS_POS, stepper.position((AxisEnum)L6470.axis_xref[cv]));
459 }
460 #endif
461 set_bed_leveling_enabled(true)
462 }
463
```

以上为自动调平设置

5.2.3 驱动模式设置

驱动模式设置分为 3 种：普通模式，UART 模式和 SPI 模式。

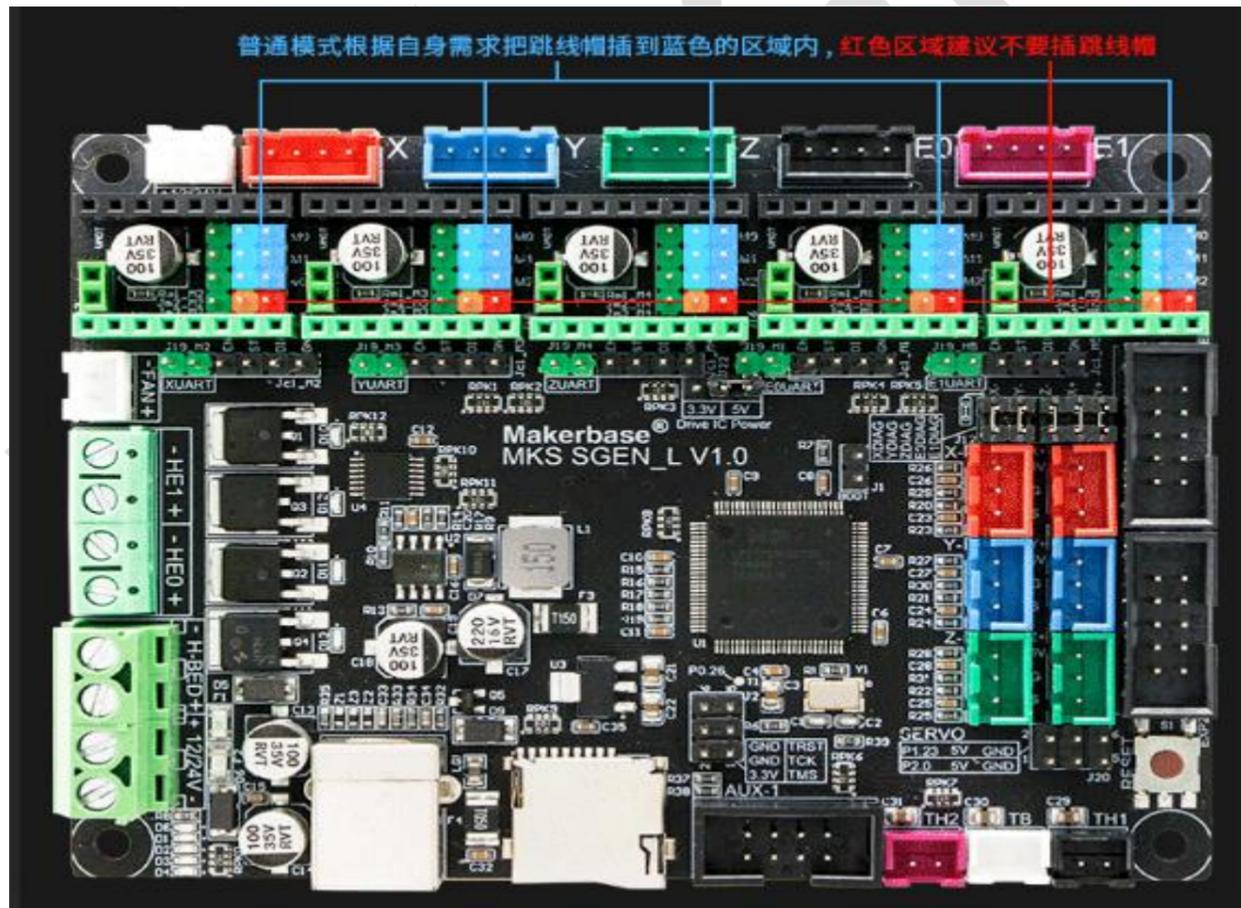
注意：以下驱动设置均基于 mks 系列驱动进行，不同厂家驱动设置方法可能设置方法不同。

普通模式（step/dir 模式）

在普通模式的情况下，基本所有驱动都是适用。

通过跳线帽调节驱动的细分。可以根据自身需求，将跳线帽插到右侧两排黑色排针的 M0,M1,M2 处，进而实现对细分的调节。

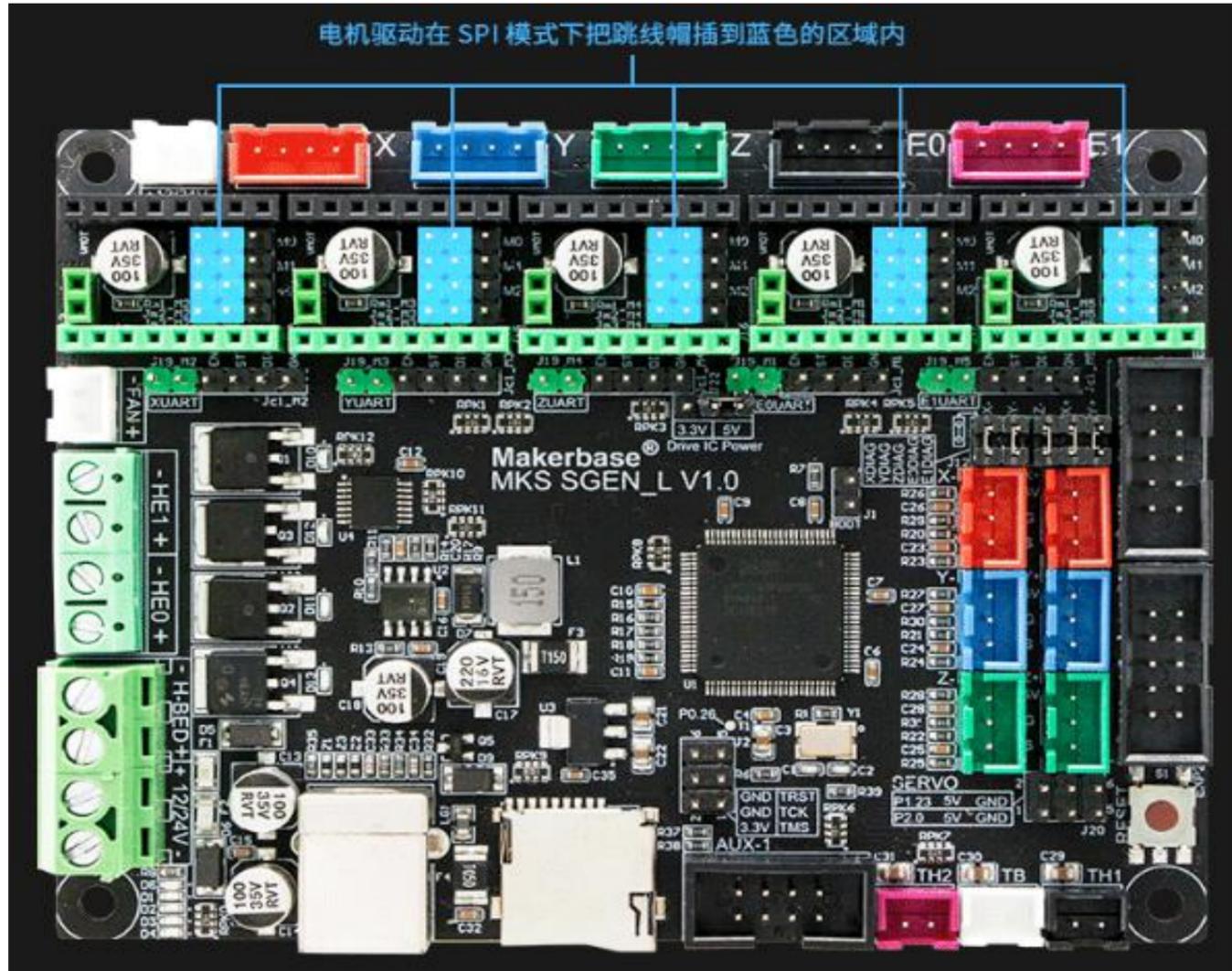
注意：在普通的模式下第四个插针，建议不带跳线帽。但是如果戴上跳线帽可能会影响部分驱动的使用。（如 TB67S109）



TMC2130 SPI 模式设置

TMC2130 SPI 模式设置分为硬件设置和软件设置

硬件设置为，主板跳线帽设置，只需要根据要求插好跳线帽，不需要进行飞线。如图所示只需要将左侧四排跳线帽插上即可



软件设置：需要 marlin2.0 固件当中设置相应的 SPI 参数，设置步骤如下

1. 驱动类型设置位 TMC2130

```

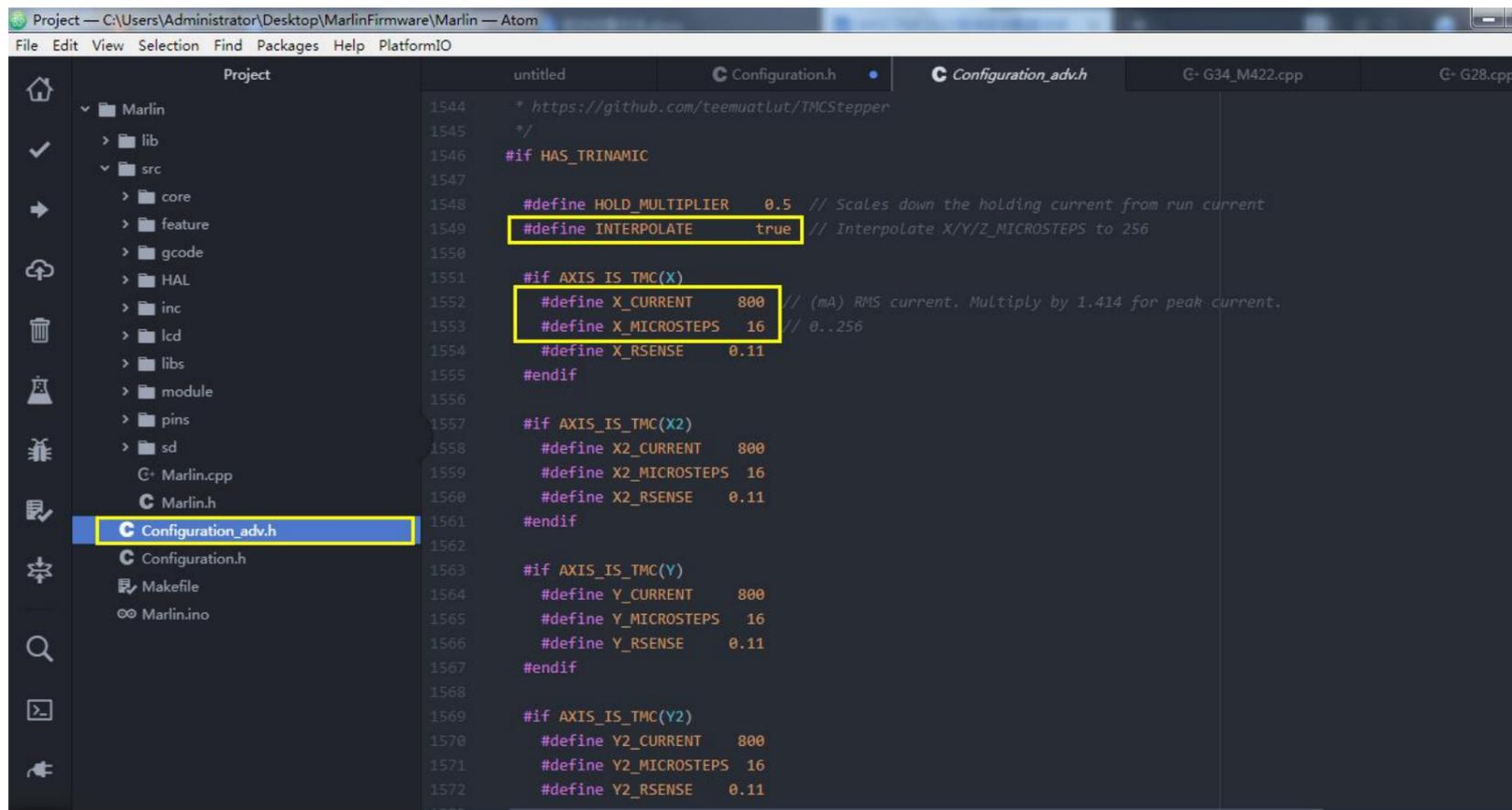
Project — C:\Users\Administrator\Desktop\MarlinFirmware\Marlin — Atom
File Edit View Selection Find Packages Help PlatformIO

Project: Marlin
  lib
  src
    core
    feature
    gcode
    HAL
    inc
    lcd
    libs
    module
    pins
    sd
  G+ 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', 'TMC2208', 'TMC2208_STANDALONE', 'TMC26X', 'TMC26X_STANDALONE', 'TMC2660', 'TMC2660_STANDALONE', 'TMC5130', 'TMC5130_STANDALONE', 'TMC5160', 'TMC5160_STANDALONE']
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. 在 Configuration_adv.h 文件中，进行配置细分和电流。



```
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 #endif
```

细分设置: X_MICROSTEPS, 一般设置为 16 细分, 因为 使能 #define INTERPOLATE true 所以芯片内部会将 16 细分拓展为 256 细分

电流设置: #define X_CURRENT 后面的为电流值, TMC2130 驱动电流范围为 **500-600mA**。高于 600mA, 在没有较好散热条件的发热比较严重。如果散热条件较好的情况下 (散热片加风扇散热) 可超过 600mA, 但最大值不超过 **1000mA**。温度太高会影响芯片的运行, 可能会造成丢步的情况。

实际驱动的的电流值, 为设置的 **1.414** 倍, 所以设置的时候需要注意不要超过驱动或者电机的最大电流

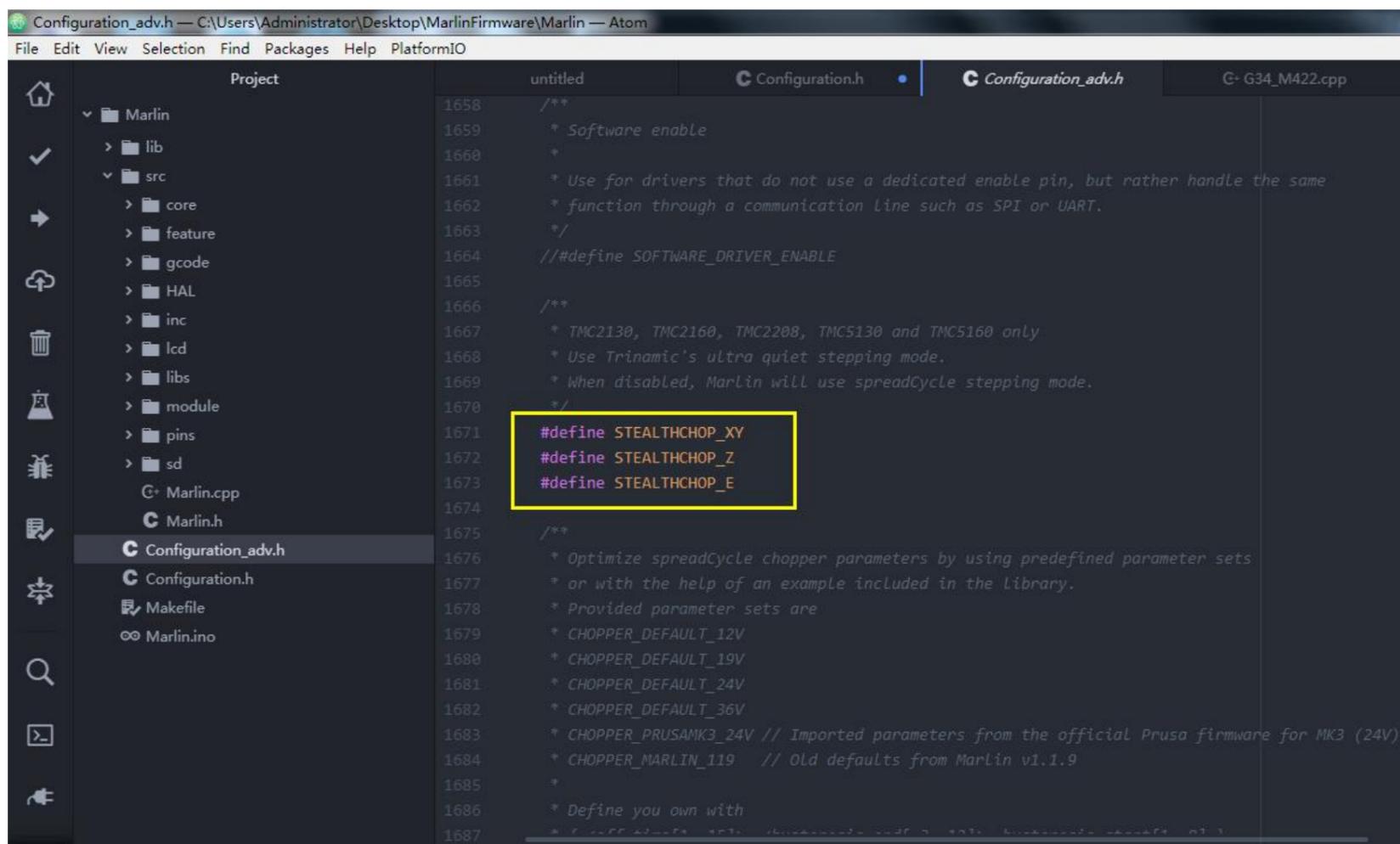
例如: 图片在设置位 **800mA**, 实际运行的电流 $I=1.414*800mA$ 。

3. 运行模式选择

SPI 模式将采用软件 IO 模拟 SPI 方式, 不需要对固件里面的硬件 SPI 进行定义默认即可。

在 Configuration_adv.h 文件进行工作模式配置

使能 STEALTHCHOP 时为静音驱动模式, 如果注释掉 STEALTHCHOP (在该项前加双斜线就可”) 则为 spreadCycle 平滑运行模式



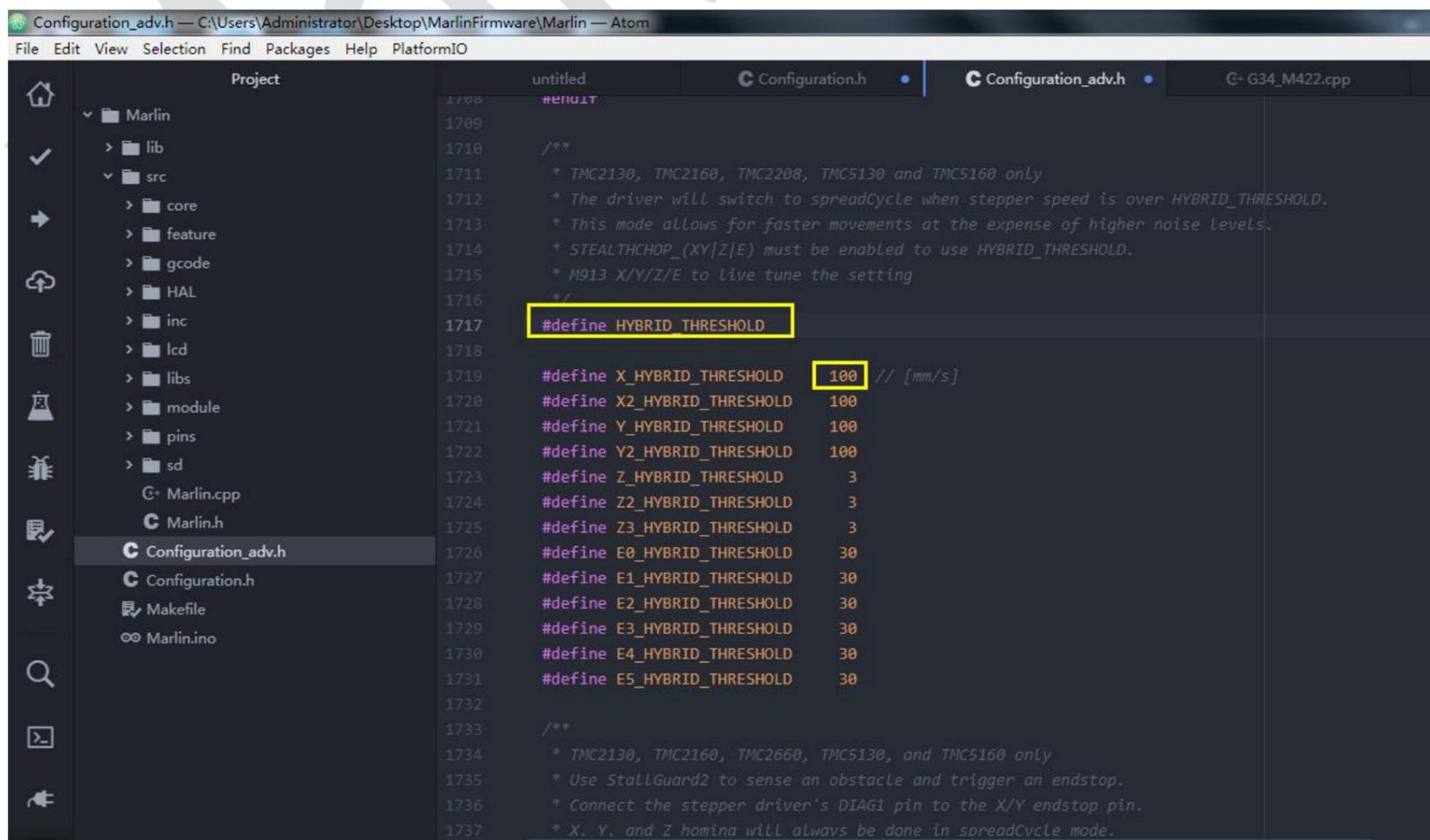
```
1658 /**
1659  * Software enable
1660  *
1661  * Use for drivers that do not use a dedicated enable pin, but rather handle the same
1662  * function through a communication line such as SPI or UART.
1663  */
1664 // #define SOFTWARE_DRIVER_ENABLE
1665
1666 /**
1667  * TMC2130, TMC2160, TMC2208, TMC5130 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 you own with
1687  * #define CHOPPER_DEFAULT_12V
```

如果要配置混合模式，需先使能 STEALTHCHOP 静音模式

使能 #define HYBRID_THRESHOLD

对应 HYBRID_THRESHOLD 项后的数值为模式切换的值。如果电机运行的速度大于设置的 100mm/s，则会切换到 spreadcycle 模式，低于 100mm/s 则为 stealthchop 模式

混合模式可以根据自身需求进行选择配置



```
1709 #define HYBRID_THRESHOLD
1710 /**
1711  * TMC2130, TMC2160, TMC2208, TMC5130 and TMC5160 only
1712  * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
1713  * This mode allows for faster movements at the expense of higher noise levels.
1714  * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
1715  * M913 X/Y/Z/E to live tune the setting
1716  */
1717 #define HYBRID_THRESHOLD
1718
1719 #define X_HYBRID_THRESHOLD 100 // [mm/s]
1720 #define X2_HYBRID_THRESHOLD 100
1721 #define Y_HYBRID_THRESHOLD 100
1722 #define Y2_HYBRID_THRESHOLD 100
1723 #define Z_HYBRID_THRESHOLD 3
1724 #define Z2_HYBRID_THRESHOLD 3
1725 #define Z3_HYBRID_THRESHOLD 3
1726 #define E0_HYBRID_THRESHOLD 30
1727 #define E1_HYBRID_THRESHOLD 30
1728 #define E2_HYBRID_THRESHOLD 30
1729 #define E3_HYBRID_THRESHOLD 30
1730 #define E4_HYBRID_THRESHOLD 30
1731 #define E5_HYBRID_THRESHOLD 30
1732
1733 /**
1734  * TMC2130, TMC2160, TMC2660, TMC5130, and TMC5160 only
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.
```

4. CS 引脚设置

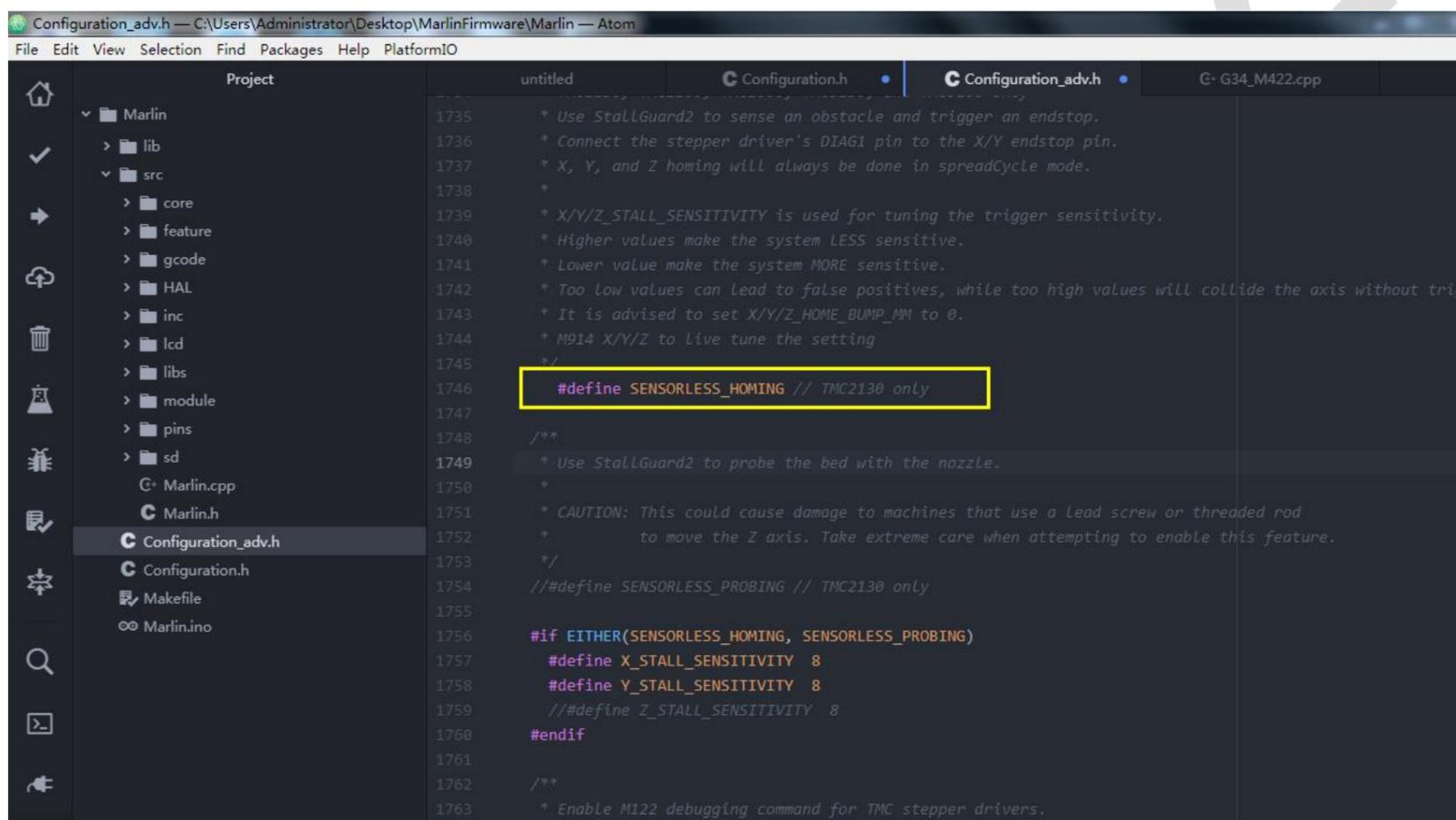
因为主板内部线路集成，使用 SGEN-L 主板，板卡文件 pin 已设置完成。默认设置即可，不用进行修改。

5. 无限位回零设置

无限位回零配置（调试难度大，可选配）

若使用普通机械限位，可以不用配置此项

在 Configuration_adv.h 文件，使能 #define SENSORLESS_HOMING



```

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 value make the system MORE sensitive.
1742  * Too low values can lead to false positives, while too high values will collide the axis without trig
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  /**
1749  * Use StallGuard2 to probe the bed with the nozzle.
1750  *
1751  * CAUTION: This could cause damage to machines that use a lead screw or threaded rod
1752  * to move the Z axis. Take extreme care when attempting to enable this feature.
1753  */
1754  //#define SENSORLESS_PROBING // TMC2130 only
1755
1756  #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
1757    #define X_STALL_SENSITIVITY 8
1758    #define Y_STALL_SENSITIVITY 8
1759    //#define Z_STALL_SENSITIVITY 8
1760  #endif
1761
1762  /**
1763  * Enable M122 debugging command for TMC stepper drivers.

```

注意：需要将驱动器 **DIAG1** 引脚的导线连接到 X/Y 限位引脚。（Z 轴暂时不支持）

如果使用 TMC2130 SGL 版本则不需要进行飞线。

X, Y 和 Z 归位将始终在 spreadCycle 模式下完成。

X/Y/Z_HOME_SENSITIVITY 用于调整触发灵敏度。值越高，系统灵敏性越低。较低的值使系统灵敏性更高。值太低会导致误报，而太高的值会在不触发的情况下碰撞轴。

建议将 X/Y/Z_HOME_BUMP_MM 设置为 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

```

注意：由于 TMC2130 处理的阻塞检测的逻辑信号为 true，所以当用阻塞检测作为限位时，限位电平只能设置为 true，否则编译会提示错误；马林固件 Z 轴暂时不能单用阻塞检测作为限位，用阻塞检测作为限位时也需要限位开关信号，不建议 Z 轴使用阻塞检测作为限位。

```

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 /**

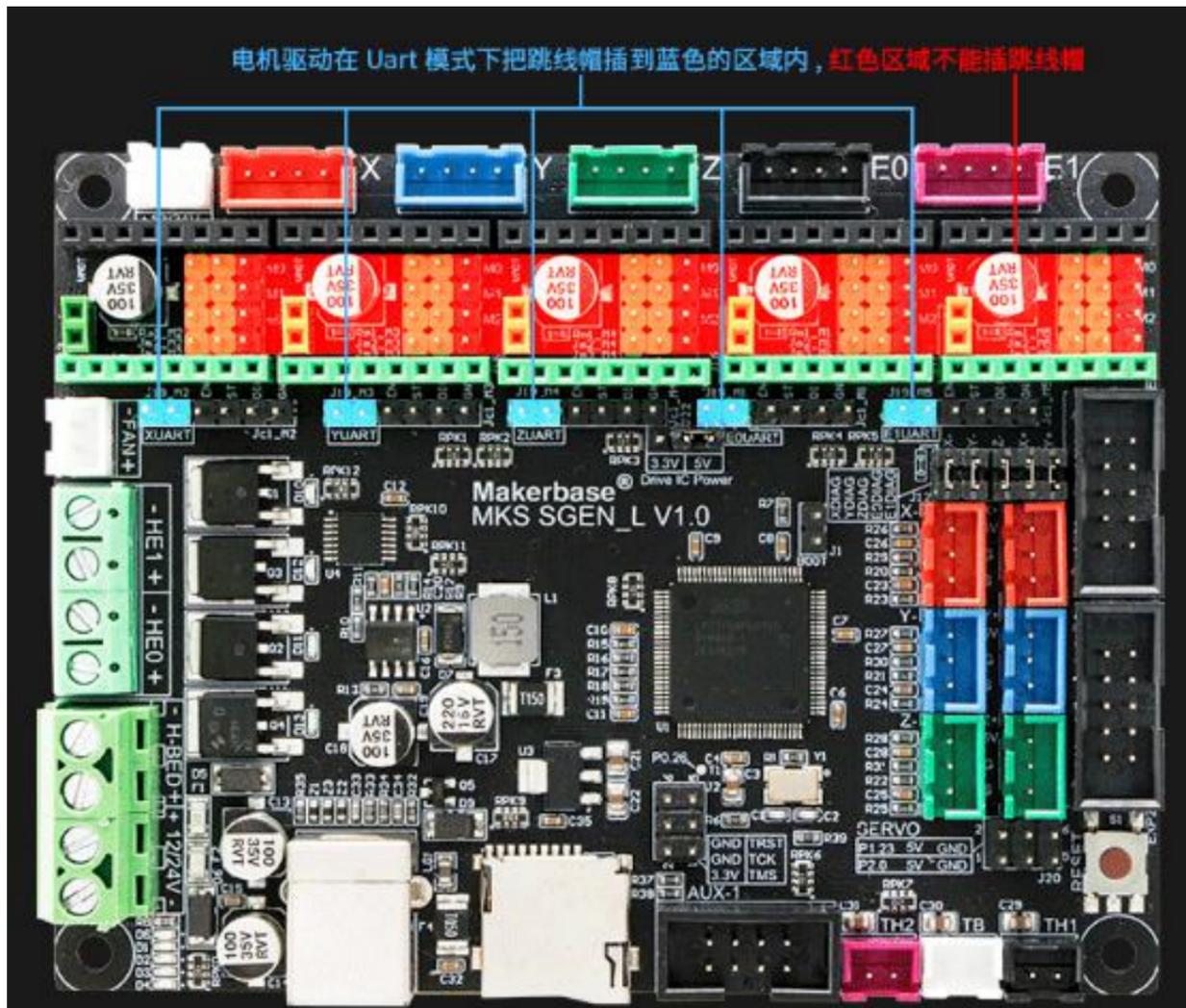
```

以上为 TMC2130 SPI 模式设置

TMC2208 UART 模式设置

SGEN-L 主板配合配合 TMC2208 使用，需要进行硬件配置和软件配置

硬件配置根据以下图片，进行跳线帽配置，使用 TMC2208 V2.0 版本，可以不用进行短接驱动上面的 NC 和 uart 端。



软件配置：UART Marlin2.0 配置方法和 2130 配置方法类似。

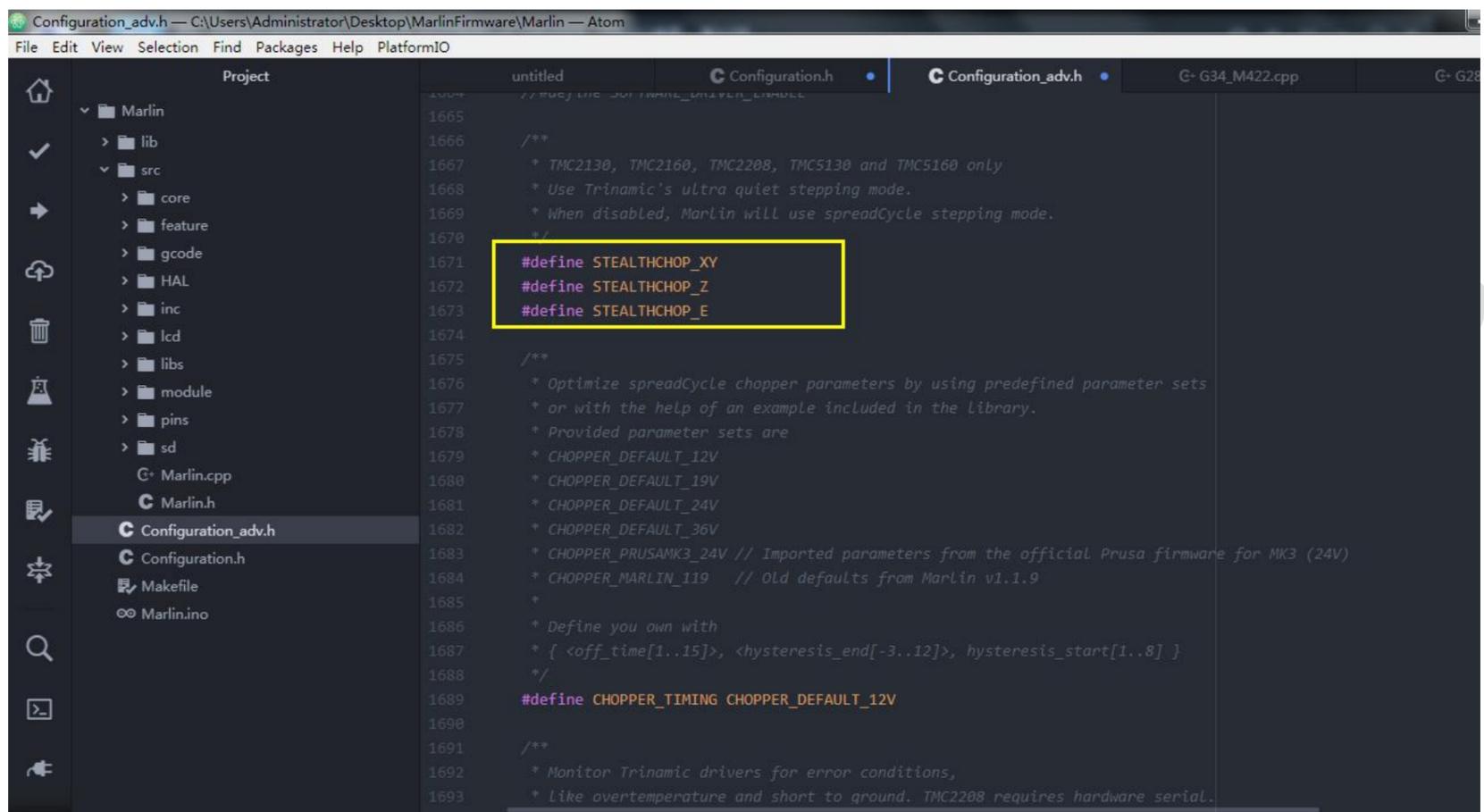
1. 设置驱动类型

```

Configuration.h — C:\Users\Administrator\Desktop\MarlinFirmware\Marlin — Atom
File Edit View Selection Find Packages Help PlatformIO
Project untitle Configuration.h Configuration_adv.h G- G34_M422.cpp
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
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 *
649 *
650 #define X_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. 运行模式选择

使能 STEALTHCHOP 时为静音驱动模式，如果注释掉 STEALTHCHOP（在该项前加双斜线就可”//）则为 spreadCycle 平滑运行模式



```
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 you 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.
```

如果要配置混合模式，需先使能 STEALTHCHOP 静音模式

使能 #define HYBRID_THRESHOLD

对应 HYBRID_THRESHOLD 项后的数值为模式切换的值。如果电机运行的速度大于设置的 100mm/s，则会切换到 spreadcycle 模式，低于 100mm/s 则为 stealthchop 模式

混合模式可以根据自身需求进行选择配置

```

1717 #define HYBRID_THRESHOLD 100
1718
1719 #define X_HYBRID_THRESHOLD 100 // [mm/s]
1720 #define X2_HYBRID_THRESHOLD 100
1721 #define Y_HYBRID_THRESHOLD 100
1722 #define Y2_HYBRID_THRESHOLD 100
1723 #define Z_HYBRID_THRESHOLD 3
1724 #define Z2_HYBRID_THRESHOLD 3
1725 #define Z3_HYBRID_THRESHOLD 3
1726 #define E0_HYBRID_THRESHOLD 30
1727 #define E1_HYBRID_THRESHOLD 30
1728 #define E2_HYBRID_THRESHOLD 30
1729 #define E3_HYBRID_THRESHOLD 30
1730 #define E4_HYBRID_THRESHOLD 30
1731 #define E5_HYBRID_THRESHOLD 30

```

3. 驱动电流设置

```

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 #endif

```

细分设置: X_MICROSTEPS, 一般设置为 16 细分, 因为 使能 #define INTERPOLATE true 所以芯片内部会将 16 细分拓展为 256 细分

实际驱动的的电流值, 为设置的 1.414 倍, 所以设置的时候需要注意不要超过驱动或者电机的最大电流

4. UART (RX 与 TX)引脚设置

因为主板内部线路集成，使用 SGEN-L 主板，板卡文件 pin 已设置完成。默认设置即可，不用进行修改。

以上为驱动部分设置的全部内容

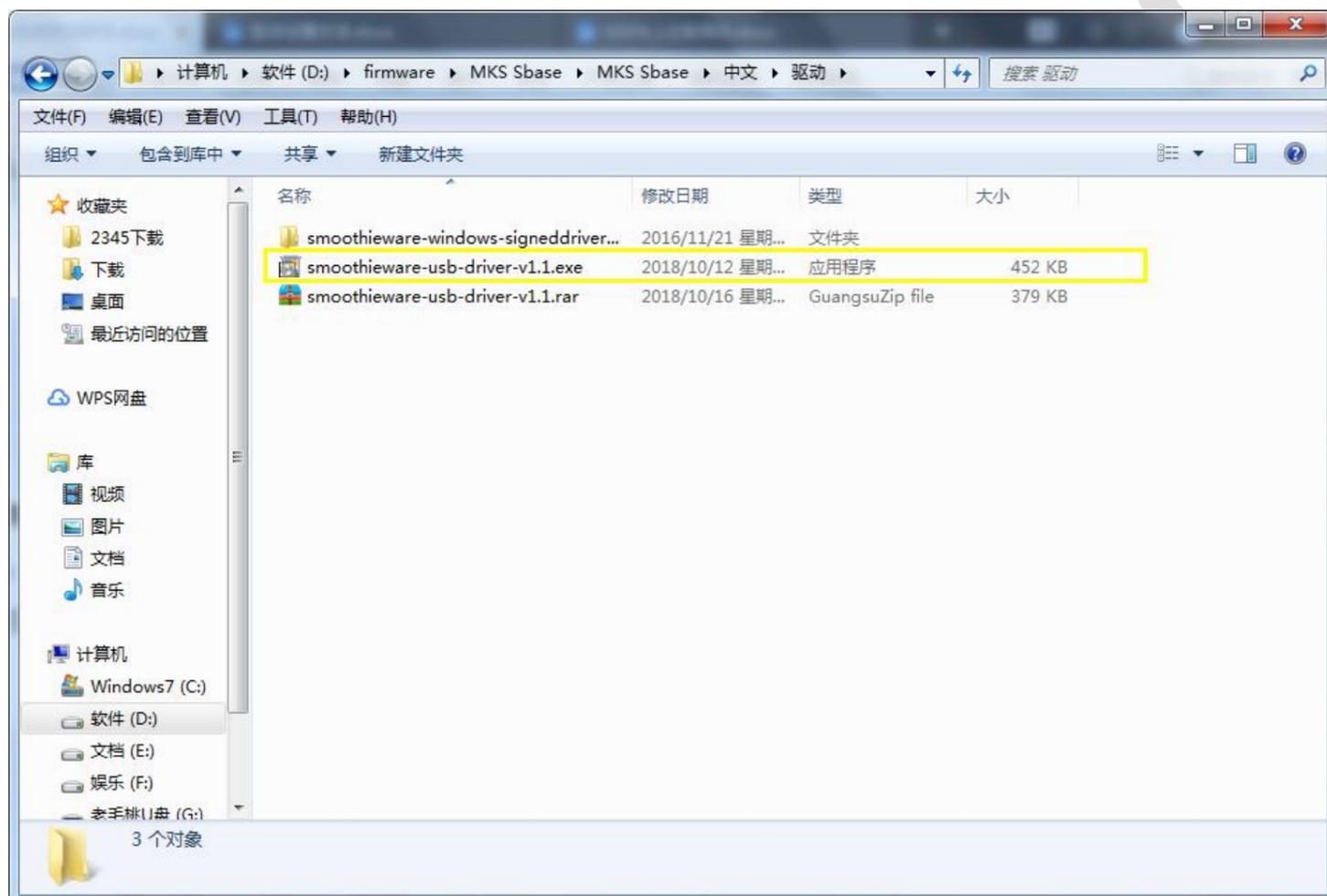
Makerbase

六. 连接打印与驱动安装

如果需要需要联机打印，则需要在电脑安装相应的 usb 驱动，COM 口才能进行识别。

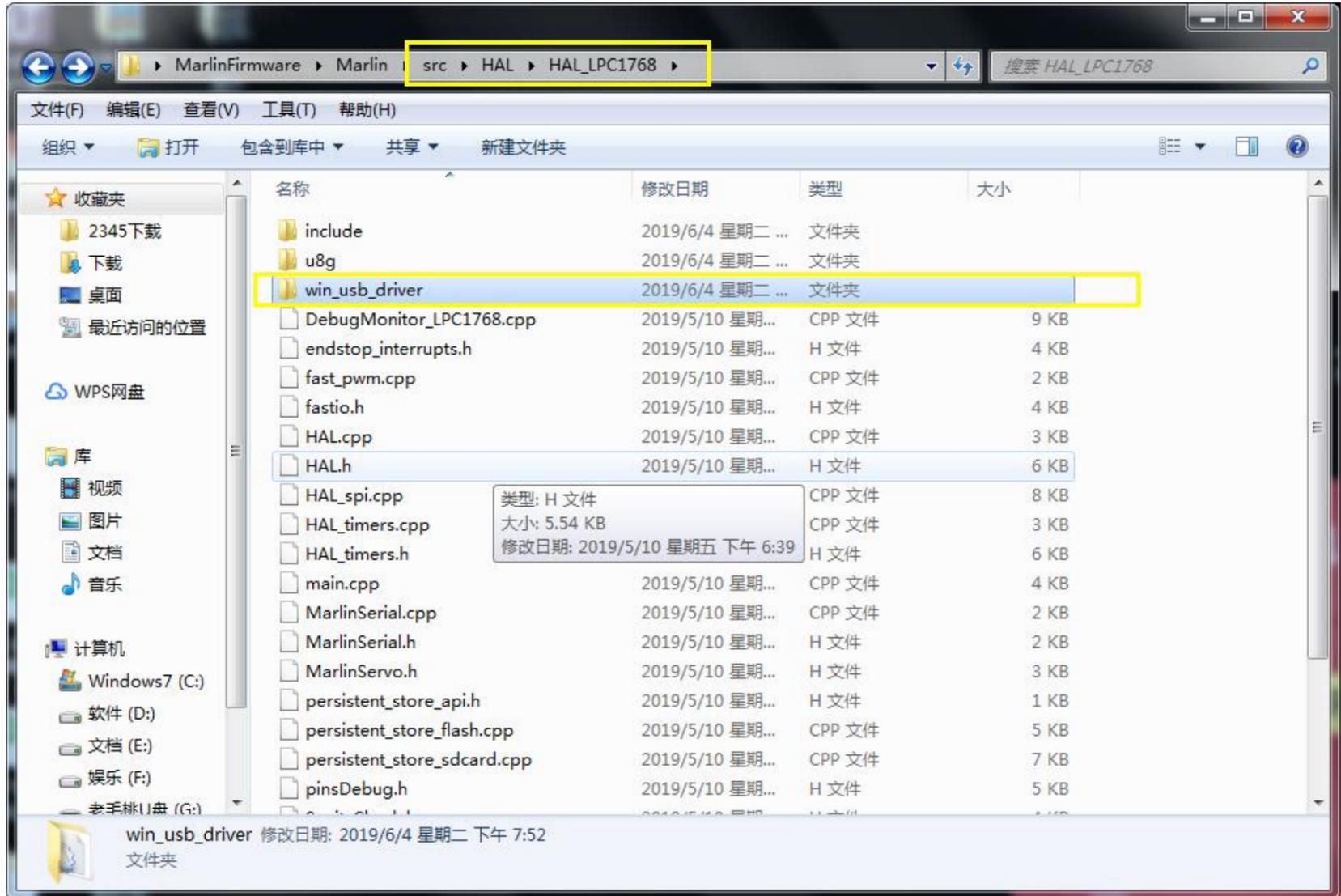
注意：主板 com 口能够被电脑识别的前提是主板已经刷新了固件，如果没有刷新固件，就算安装对应固件的 USB 串口，也无法被识别。

Smoothieware 固件的 usb 驱动，可以到 smoothieware 的官网进行下载安装，或者我们提供的固件压缩包也有包含，可以自行安装。



MARLIN2.0 固件

如果更新了 marlin 固件，但是电脑识别为未知设备的话，可以右击选择更新驱动程序，选择一下路径的进行更新驱动程序



Makerbase

七. 技术支持及保证

1. 发货前会做通电测试，保证可以正式使用才发货。
2. 欢迎各位朋友加入讨论群：232237692
3. 欢迎光临博客交流：<http://flyway97.blog.163.com>
4. 3D打印机主板定制，联系黄生：13148932315 谭生：15521395023 彭生：13427595835
5. 有问题可联系我们客服或者在群里找技术支持人员，我们将竭诚为您服务



创客基地官网



创客基地淘宝