

MKS Robin lite3 Datasheet

Features

STM32F103 main chip, 72MHZ main frequency, faster processing speed than ATMEGA2560.

Specially running Marlin V2.0 firmware.

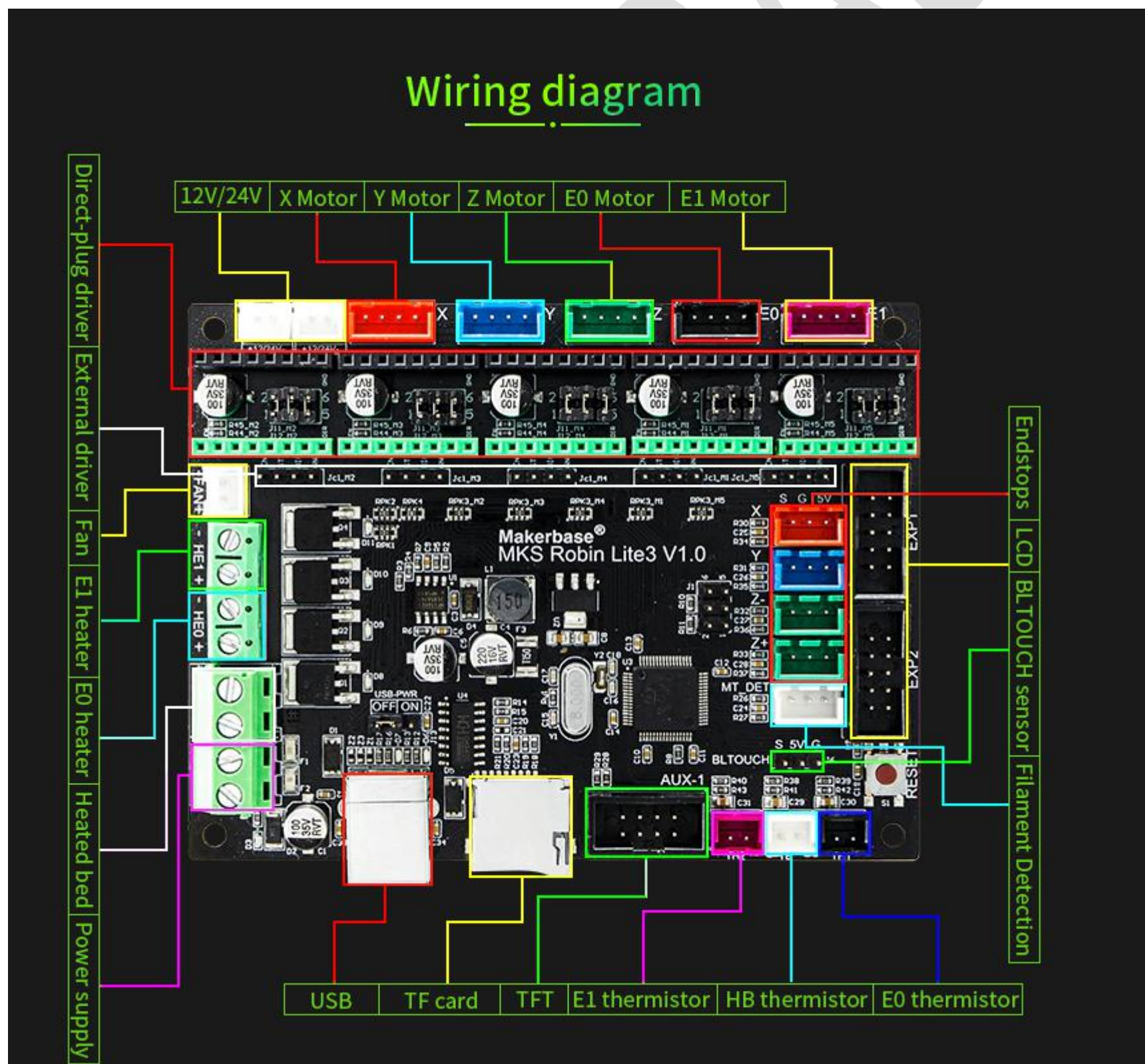
Compatible with different LCD, MKS TFT and MKS PAD.

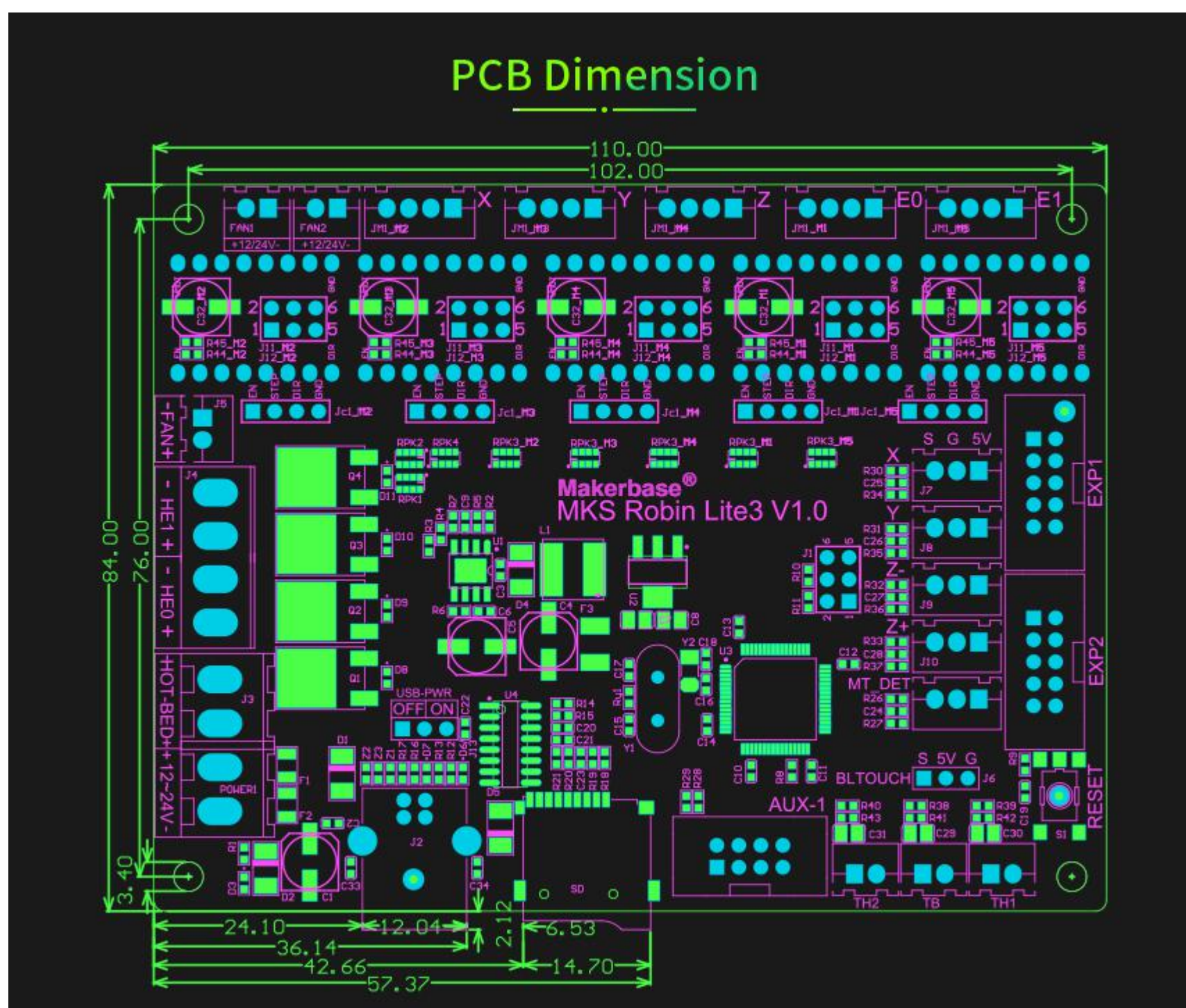
Support for both direct-plug stepper drivers and external high-current stepper drivers.

Support for auto-leveling. Reserved BLTOUCH connector.

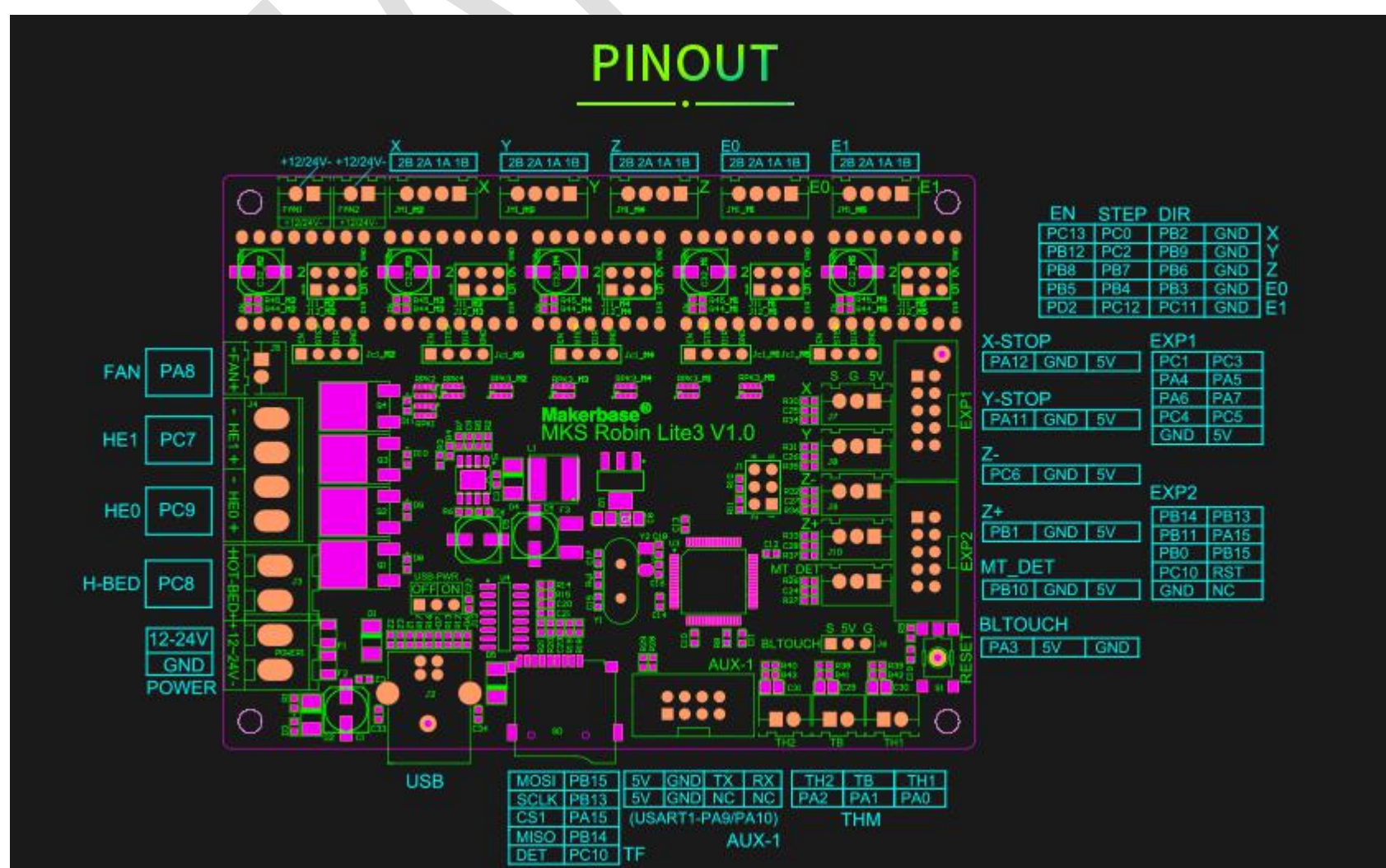
Add USB power jumper setting, reduce USB chip burnout problems.

Wiring





PINOUT



When you get MKS Robin lite3, how do you use it?

1. If you want online printing, please install USB Drive: CH340.

https://github.com/makerbase-mks/Driver/blob/master/CH340G_USB%E9%A9%B1%E5%8A%A8.rar

2. Download Marlin 2.0 firmware.

<https://github.com/makerbase-mks/MKS-Robin-lite3>

<http://marlinfw.org/meta/download/>

3. Update Marlin 2.0 firmware

Build compilation environment:ATOM+PLATFORMIO. Specific methods can refer to the following links:

[http://marlinfw.org/docs/basics/install_platformio.html#installing-marlin-\(platformio\)](http://marlinfw.org/docs/basics/install_platformio.html#installing-marlin-(platformio))

<http://docs.platformio.org/en/latest/ide/atom.html#installation>

After compiling, there will be Robin lite3.bin firmware in the folder(folder path:..pioenvs\mks_robin lite3), then follow the steps below to update the firmware :

Step 1, copy Robin.bin to SD card

Step 2, insert SD card to MKS Robin board

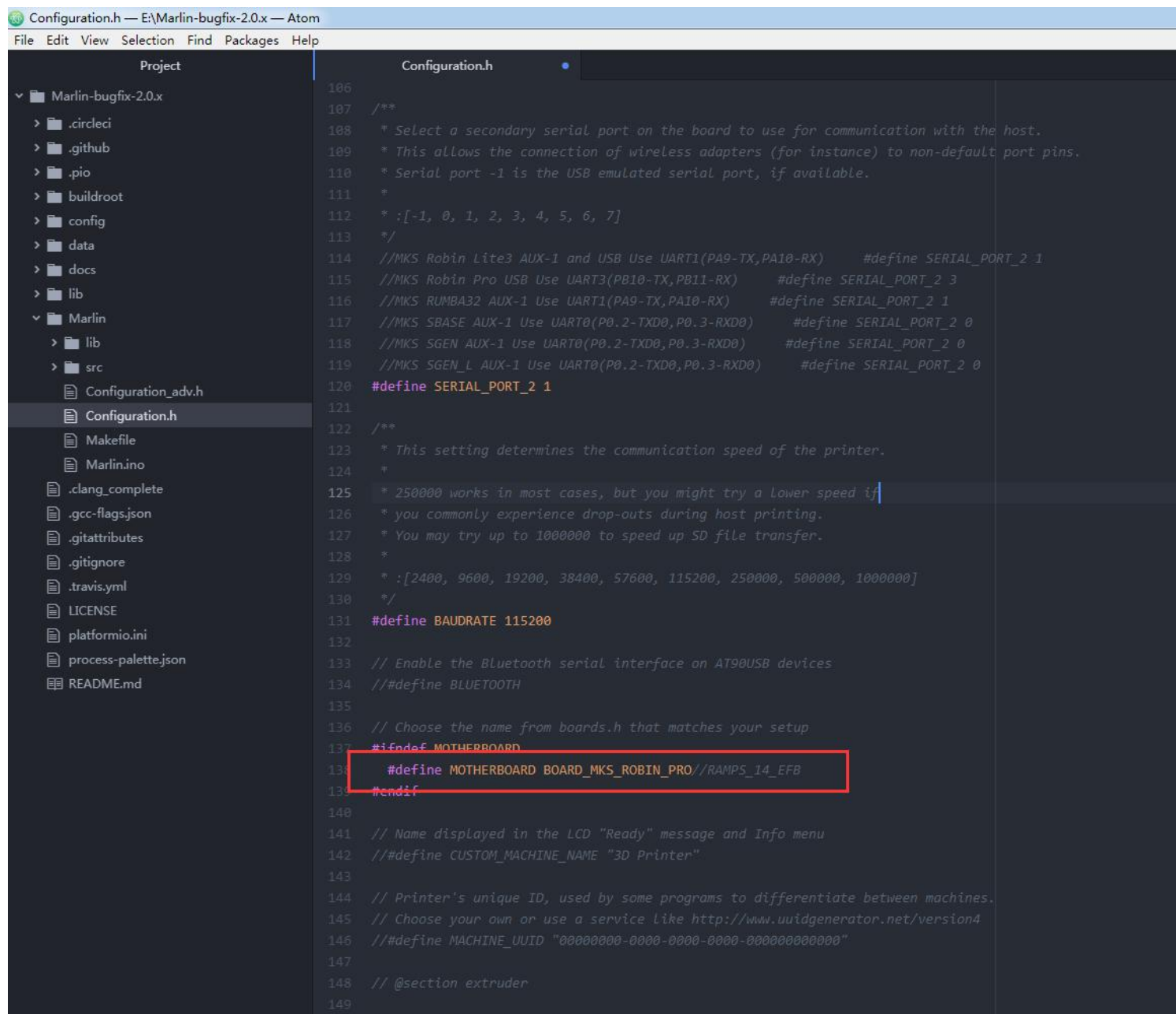
Step 3, reboot MKS Robin board

Step 4, wait until firmware upgrade is completed.

Basic parameter setting instructions

1. BAUD RATE

2. Motherboard: Board_MKS_ROBIN_LITE3

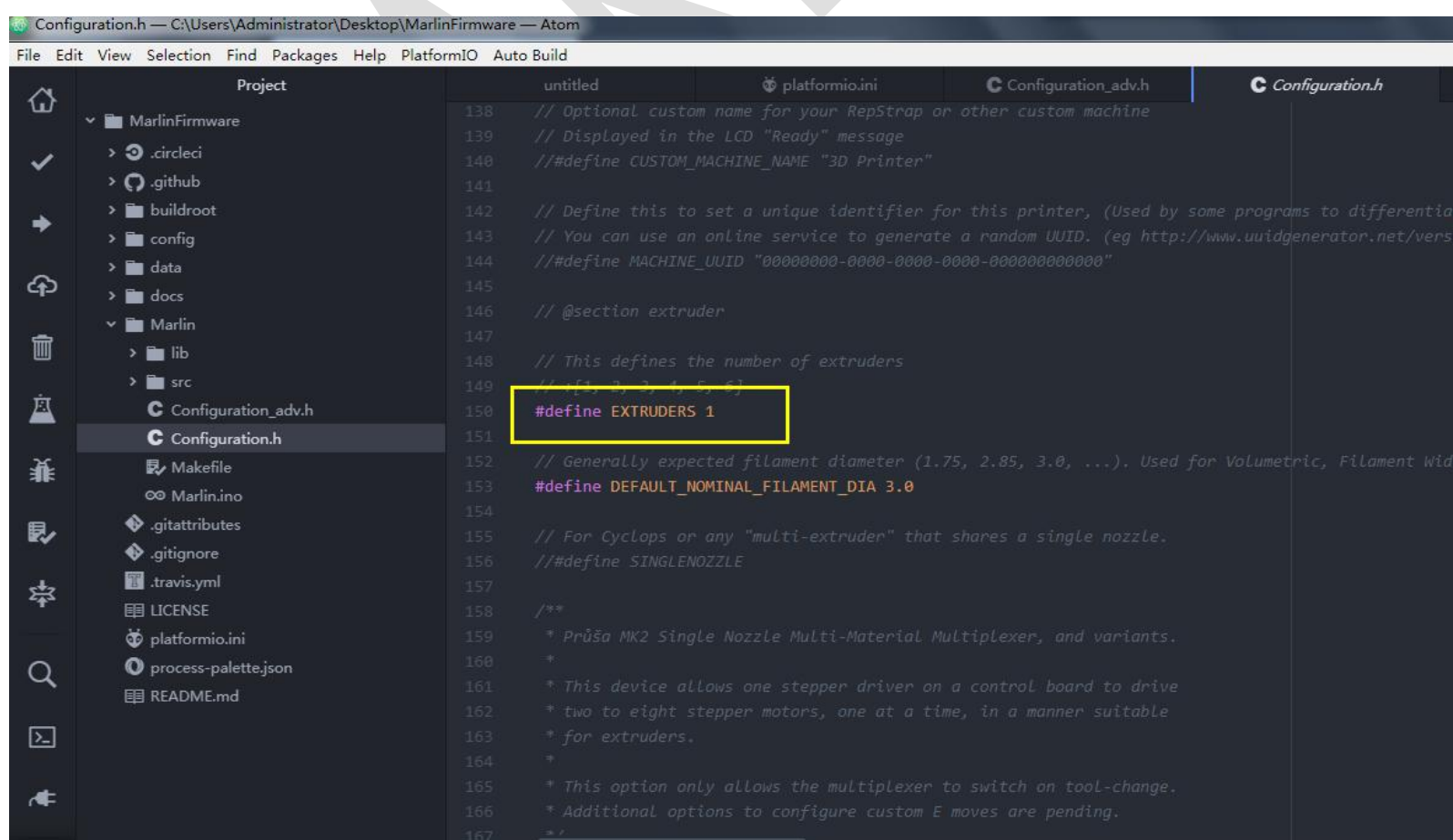


```

106
107 /**
108  * Select a secondary serial port on the board to use for communication with the host.
109  * This allows the connection of wireless adapters (for instance) to non-default port pins.
110  * Serial port -1 is the USB emulated serial port, if available.
111  *
112  * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
113  */
114 //MKS Robin Lite3 AUX-1 and USB Use UART1(PA9-TX,PA10-RX)    #define SERIAL_PORT_2 1
115 //MKS Robin Pro USB Use UART3(PB10-TX,PB11-RX)    #define SERIAL_PORT_2 3
116 //MKS RUMBA32 AUX-1 Use UART1(PA9-TX,PA10-RX)    #define SERIAL_PORT_2 1
117 //MKS SBASE AUX-1 Use UART0(P0.2-TXD0,P0.3-RXD0)    #define SERIAL_PORT_2 0
118 //MKS SGEN AUX-1 Use UART0(P0.2-TXD0,P0.3-RXD0)    #define SERIAL_PORT_2 0
119 //MKS SGEN_L AUX-1 Use UART0(P0.2-TXD0,P0.3-RXD0)    #define SERIAL_PORT_2 0
120 #define SERIAL_PORT_2 1
121
122 /**
123  * This setting determines the communication speed of the printer.
124  *
125  * 250000 works in most cases, but you might try a lower speed if
126  * you commonly experience drop-outs during host printing.
127  * You may try up to 1000000 to speed up SD file transfer.
128  *
129  * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
130  */
131 #define BAUDRATE 115200
132
133 // Enable the Bluetooth serial interface on AT90USB devices
134 // #define BLUETOOTH
135
136 // Choose the name from boards.h that matches your setup
137 #ifndef MOTHERBOARD
138 #define MOTHERBOARD BOARD_MKS_ROBIN_PRO//RAMPS_14_EFB
139 #endif
140
141 // Name displayed in the LCD "Ready" message and Info menu
142 // #define CUSTOM_MACHINE_NAME "3D Printer"
143
144 // Printer's unique ID, used by some programs to differentiate between machines.
145 // Choose your own or use a service like http://www.uuidgenerator.net/version4
146 // #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
147
148 // @section extruder
149

```

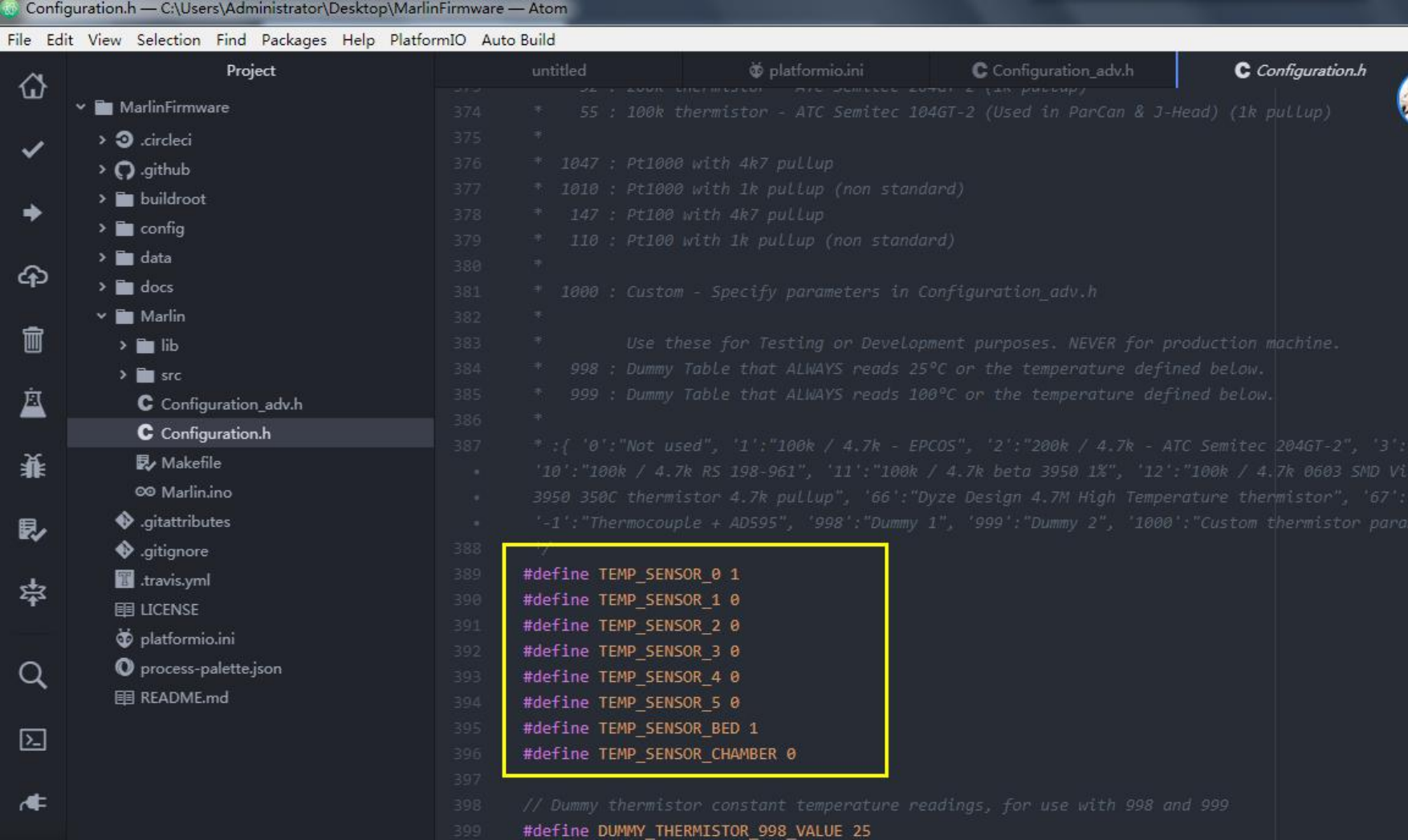
3. Extruder number and thermal type



```

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/version4)
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 Width
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':"
* '10':"100k / 4.7k RS 198-961", '11':"100k / 4.7k beta 3950 1%", '12':"100k / 4.7k 0603 SMD VL
* 3950 350C thermistor 4.7k pullup", '66':"Dyze Design 4.7M High Temperature thermistor", '67':"
* '-1':"Thermocouple + AD595", '998':"Dummy 1", '999':"Dummy 2", '1000':"Custom thermistor para
388
389 #define TEMP_SENSOR_0 1
390 #define TEMP_SENSOR_1 0
391 #define TEMP_SENSOR_2 0
392 #define TEMP_SENSOR_3 0
393 #define TEMP_SENSOR_4 0
394 #define TEMP_SENSOR_5 0
395 #define TEMP_SENSOR_BED 1
396 #define TEMP_SENSOR_CHAMBER 0
397
398 // Dummy thermistor constant temperature readings, for use with 998 and 999
399 #define DUMMY_THERMISTOR_998_VALUE 25

```

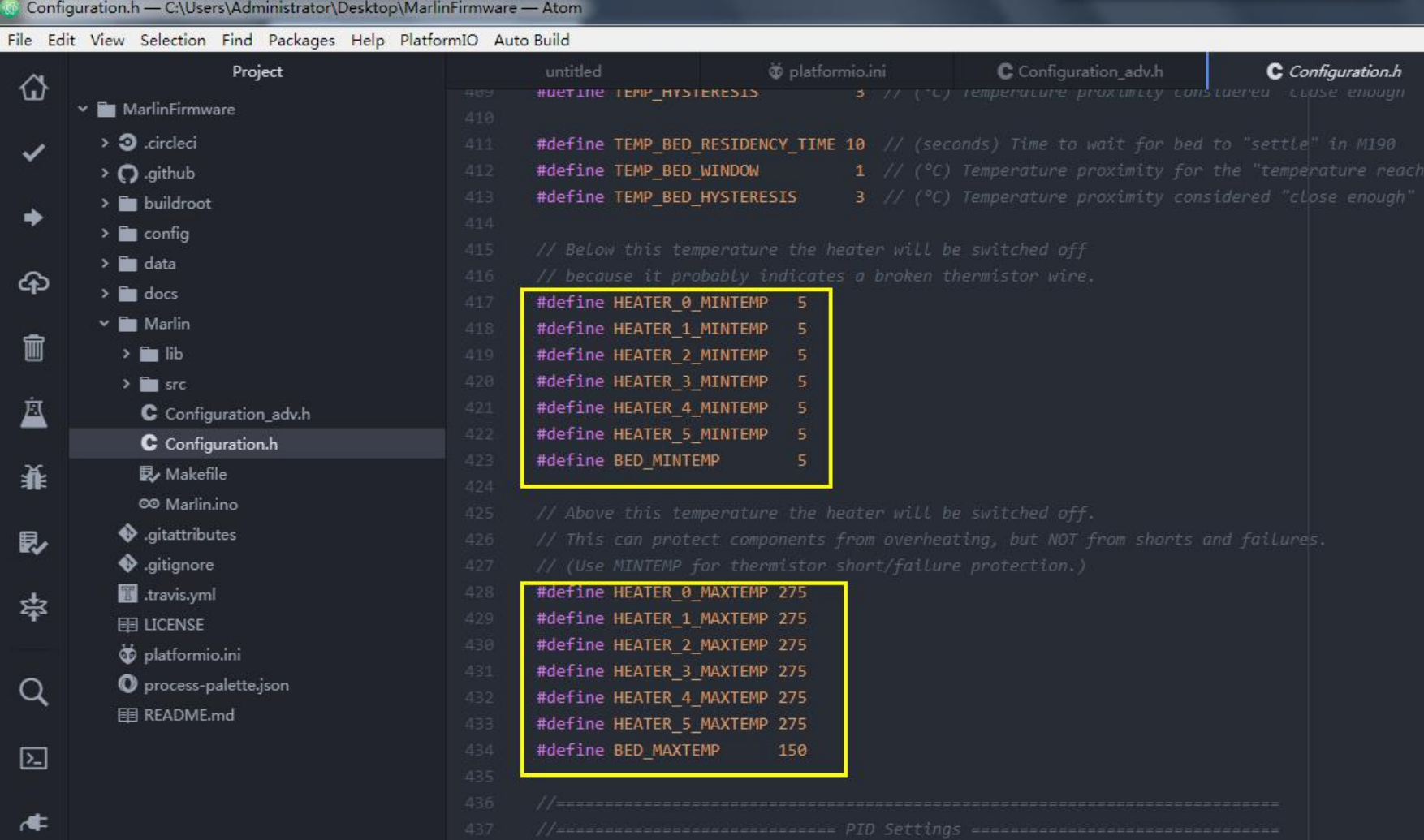
Note: Don't forget to set heated bed sensor here.

`#define TEMP_SENSOR_BED 1`(0:disable bed, 1:enable bed)

4. Temperature

1) MINTEMP and MAXTEMP.

If the actual temperature is not within this range, the motherboard will report error to the display or PC.



```

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 =====

```

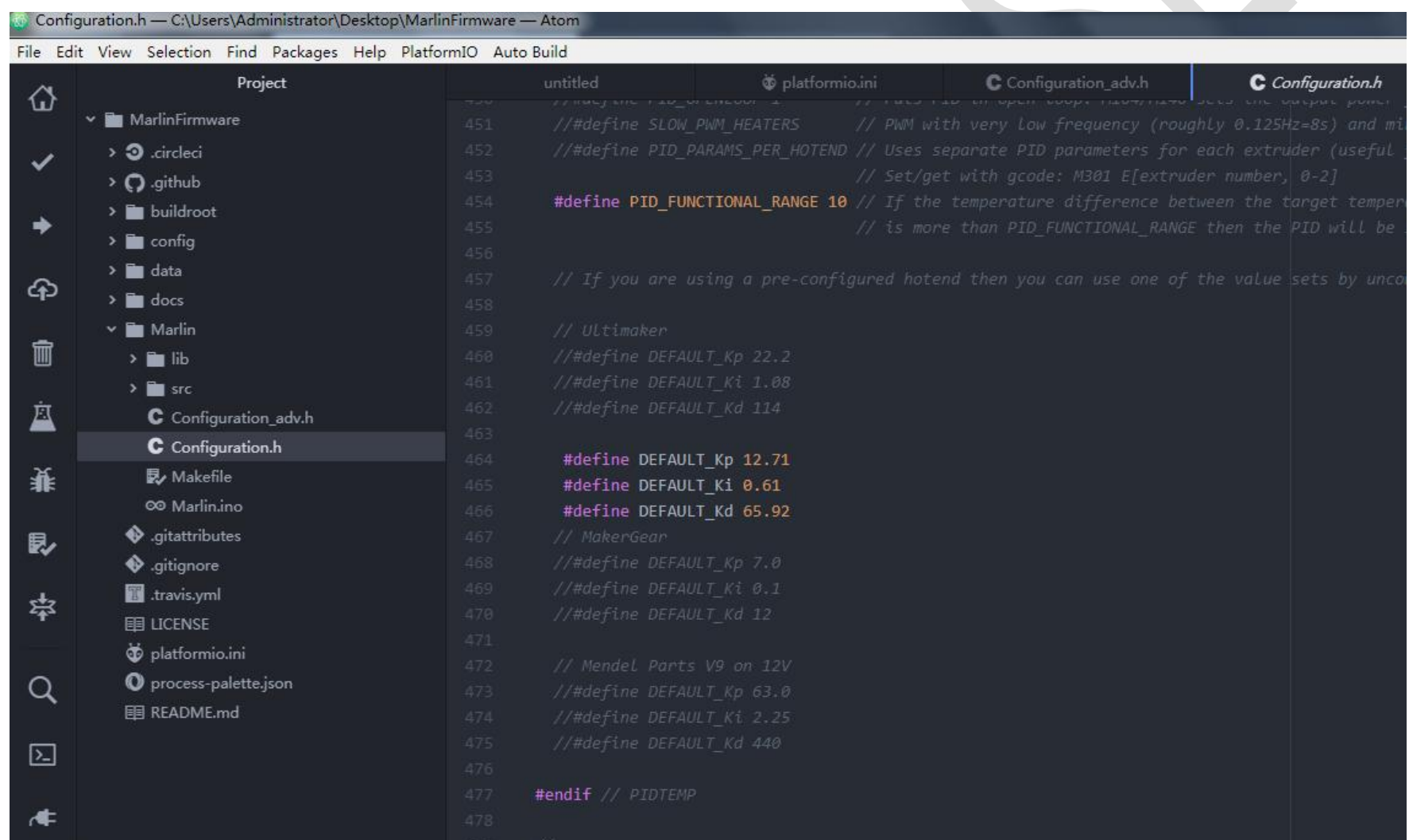

2) PID (Generally, it's not necessary to adjust it.)

Using command: M303

"M303 E0 S190"

#send this command to the pc, it will automatically run PID calculations. After the calculation is completed, fill in the returned value into the firmware.

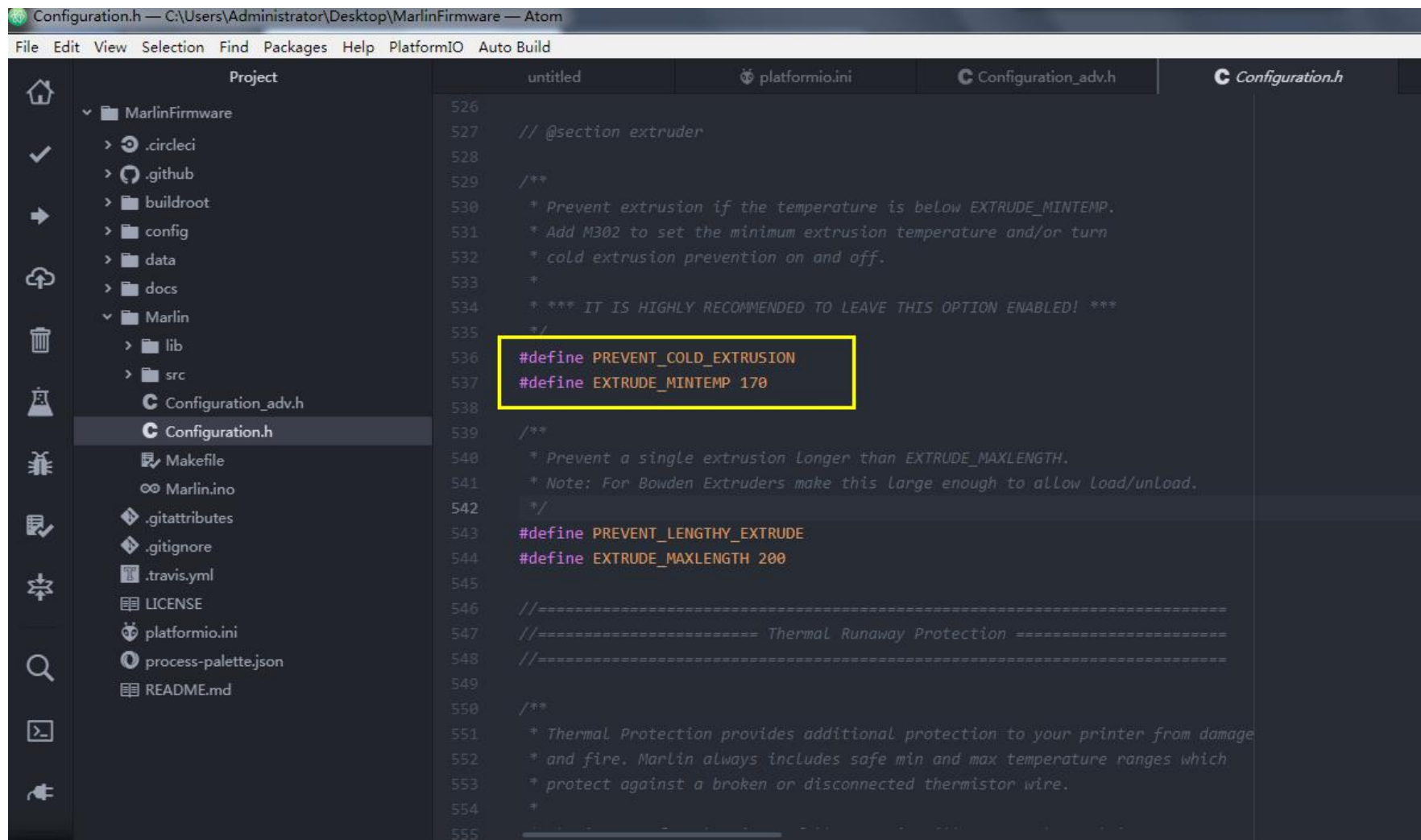
```
T: 190.4/190.0 @0 0 7/8
T: 190.2/190.0 @0 0 7/8
Cycle 7:
Max: 190.8 Min: 184.3 high time: 48.2s low time: 7.5s
Averages over last 3 cycles: Max: 81.8c Min: 79.0c high :
ku: 17.7607
tu: 23.7929
Trying:
Kp: 10.7
Ki: 0.045
Kd: 32
PID Autotune Complete! The settings above have been loaded in
```



3) PREVENT_COLD_EXTRUDER

170 degree by default. Only if the extruder temperature reaches to 170 degree, can the extruder work.

Therefore, if you don't need to use this extruder, you can lower this temperature.



```

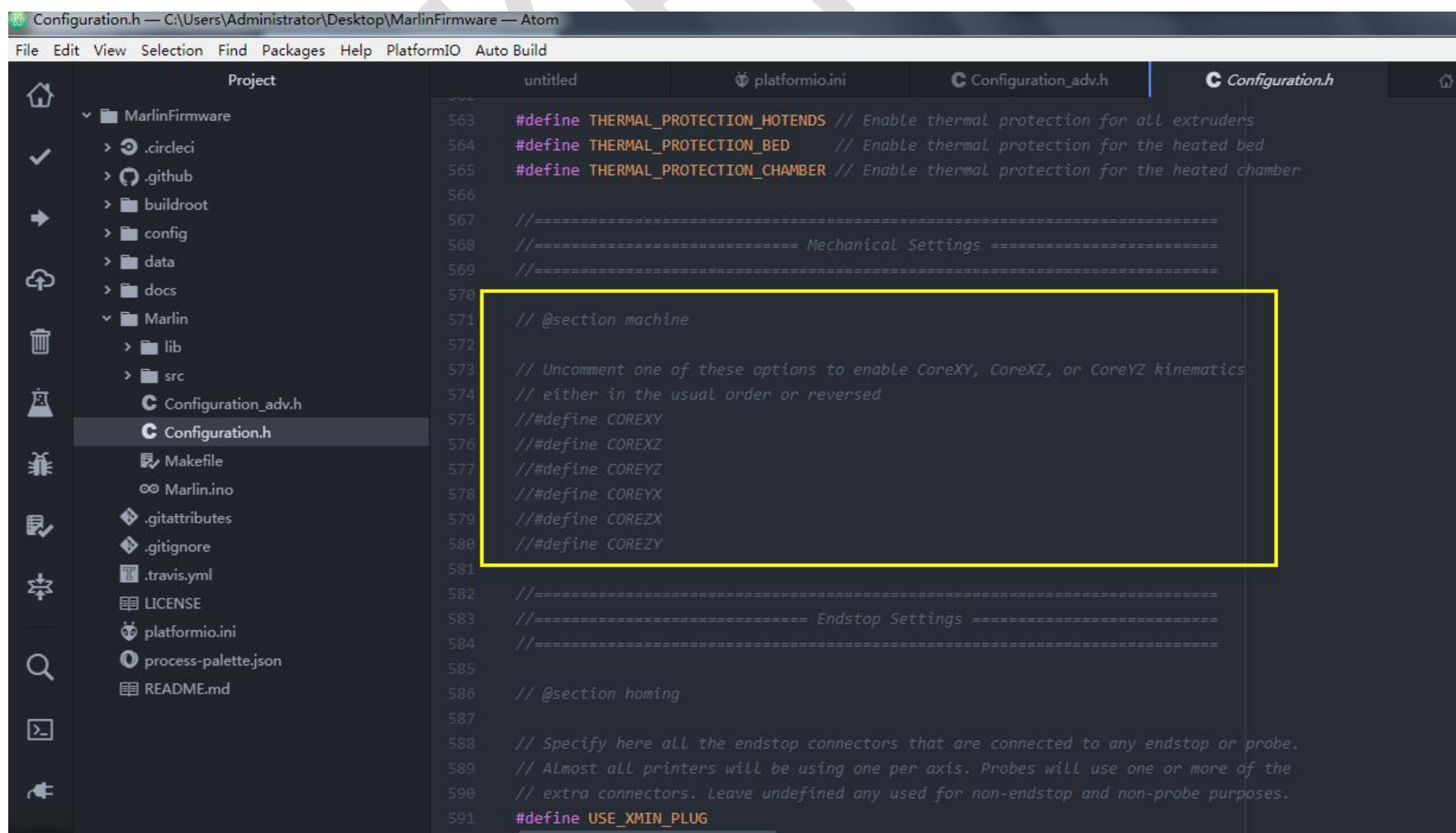
526
527 // @section extruder
528
529 /**
530  * Prevent extrusion if the temperature is below EXTRUDE_MINTEMP.
531  * Add M302 to set the minimum extrusion temperature and/or turn
532  * cold extrusion prevention on and off.
533  *
534  * *** IT IS HIGHLY RECOMMENDED TO LEAVE THIS OPTION ENABLED! ***
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 /**
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

```

5. Machine Parameters

1) Machine Setting

XYZ machine by default. You can change the machine type here as your 3d printer.

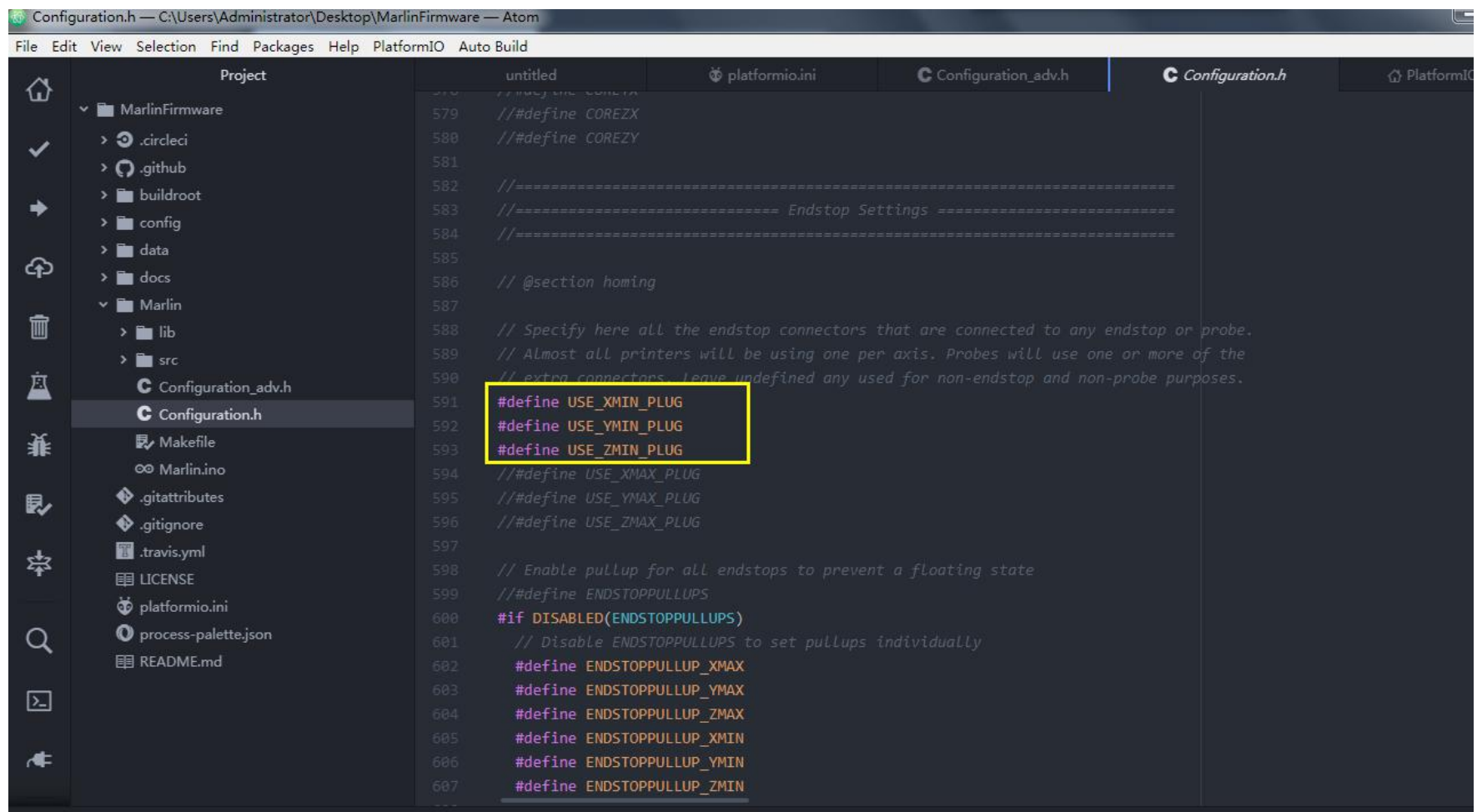


```

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 //===== Mechanical Settings =====
568 //=====
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 //===== Endstop Settings =====
583 //=====
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

```

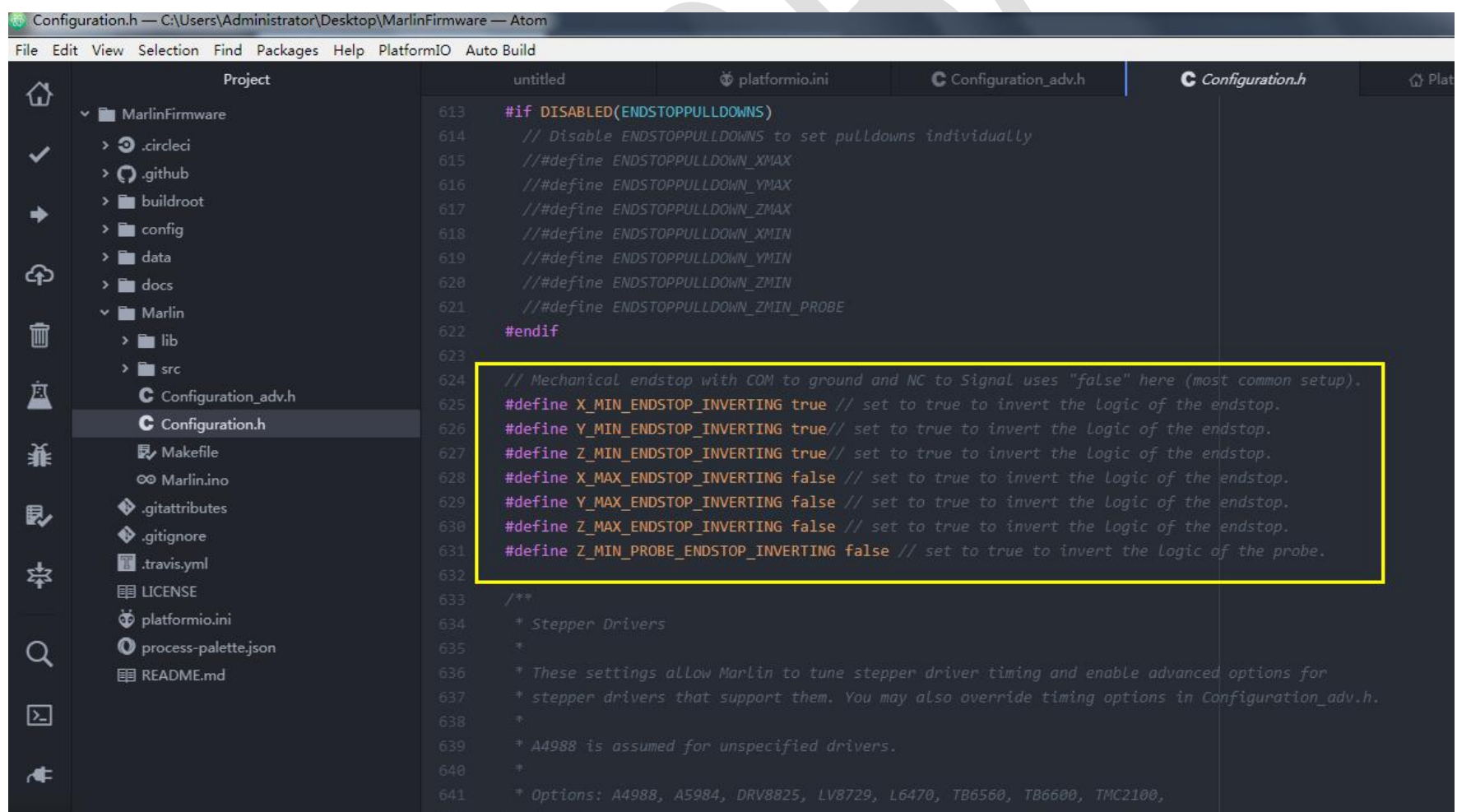
2) Endstop Setting



```

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
592 #define USE_XMIN_PLUG
593 #define USE_YMIN_PLUG
594 #define USE_ZMIN_PLUG
595
596 // #define USE_XMAX_PLUG
597 // #define USE_YMAX_PLUG
598 // #define USE_ZMAX_PLUG
599
600 // Enable pullup for all endstops to prevent a floating state
601 // #define ENDSTOPPULLUPS
602 // #if DISABLED(ENDSTOPPULLUPS)
603 // // Disable ENDSTOPPULLUPS to set pullups individually
604 // #define ENDSTOPPULLUP_XMAX
605 // #define ENDSTOPPULLUP_YMAX
606 // #define ENDSTOPPULLUP_ZMAX
607 // #define ENDSTOPPULLUP_XMIN
608 // #define ENDSTOPPULLUP_YMIN
609 // #define ENDSTOPPULLUP_ZMIN
610

```



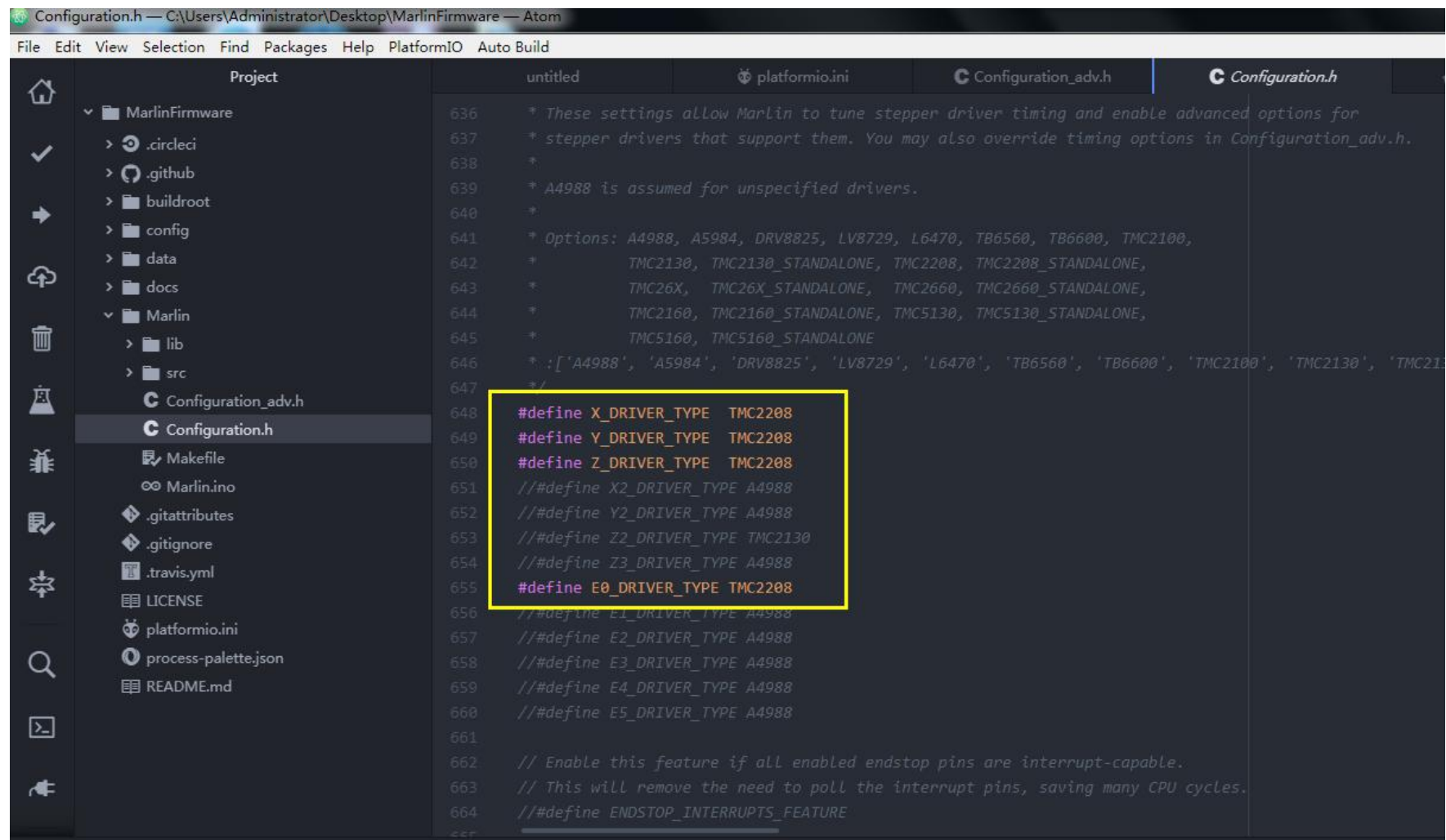
```

613 // #if DISABLED(ENDSTOPPULLDOWN)
614 // // Disable ENDSTOPPULLDOWN 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,
642

```


6. Stepper Driver Setting

Only if you set the stepper driver type, can you set the corresponding mode(TMC2208, uart mode; TMC2130, spi mode).



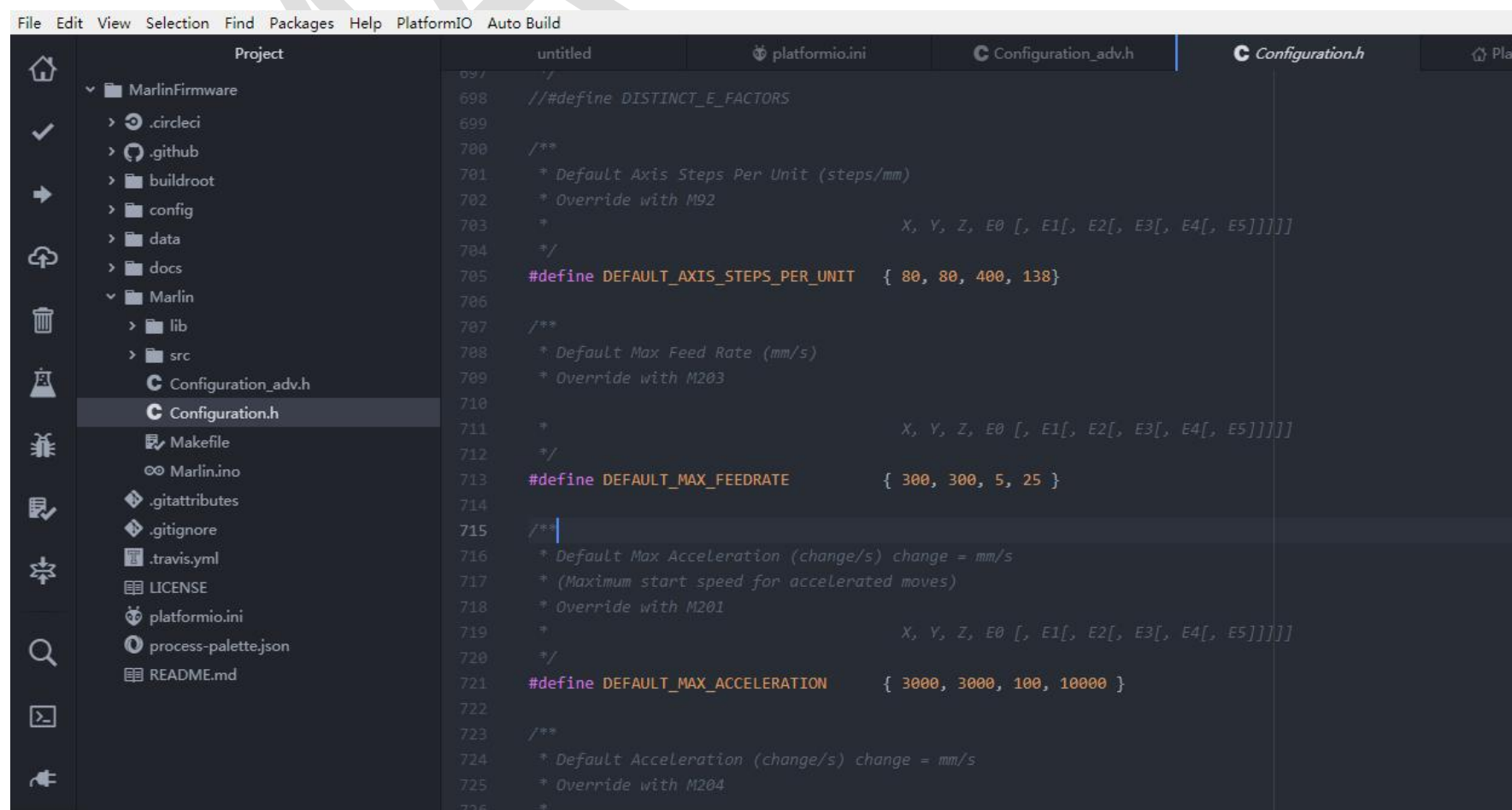
7. Stepper Motor Setting

1) Motor Pulse (#define DEFAULT_AXIS_STEPS_PER_UNIT)

Formula

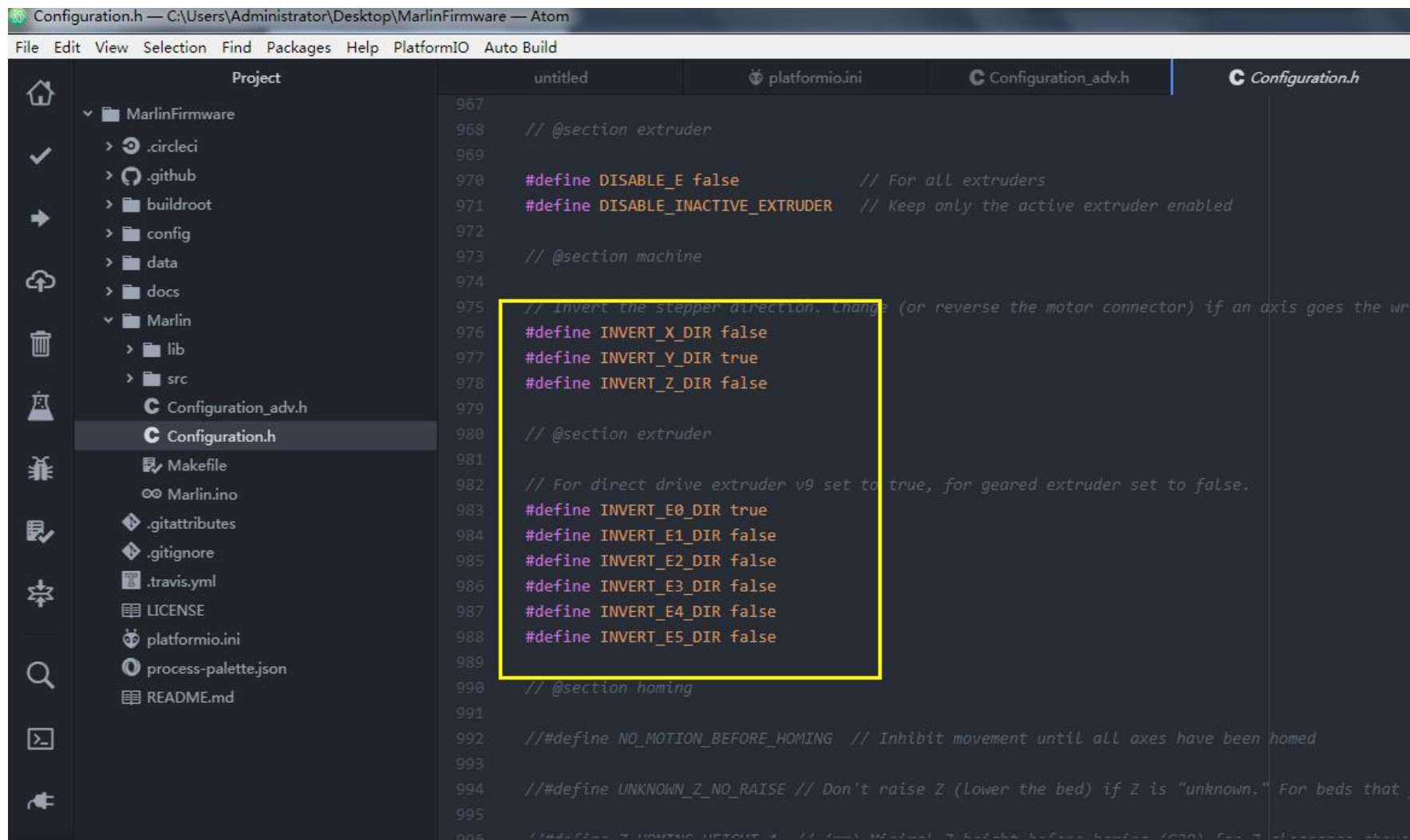
Synchronous wheel motor: $(360/\text{steps}) * \text{microstep} / (\text{diameter} * 3.14)$

Lead motor: $(360/\text{steps}) * \text{microstep} / \text{lead}$



2) Motor Direction

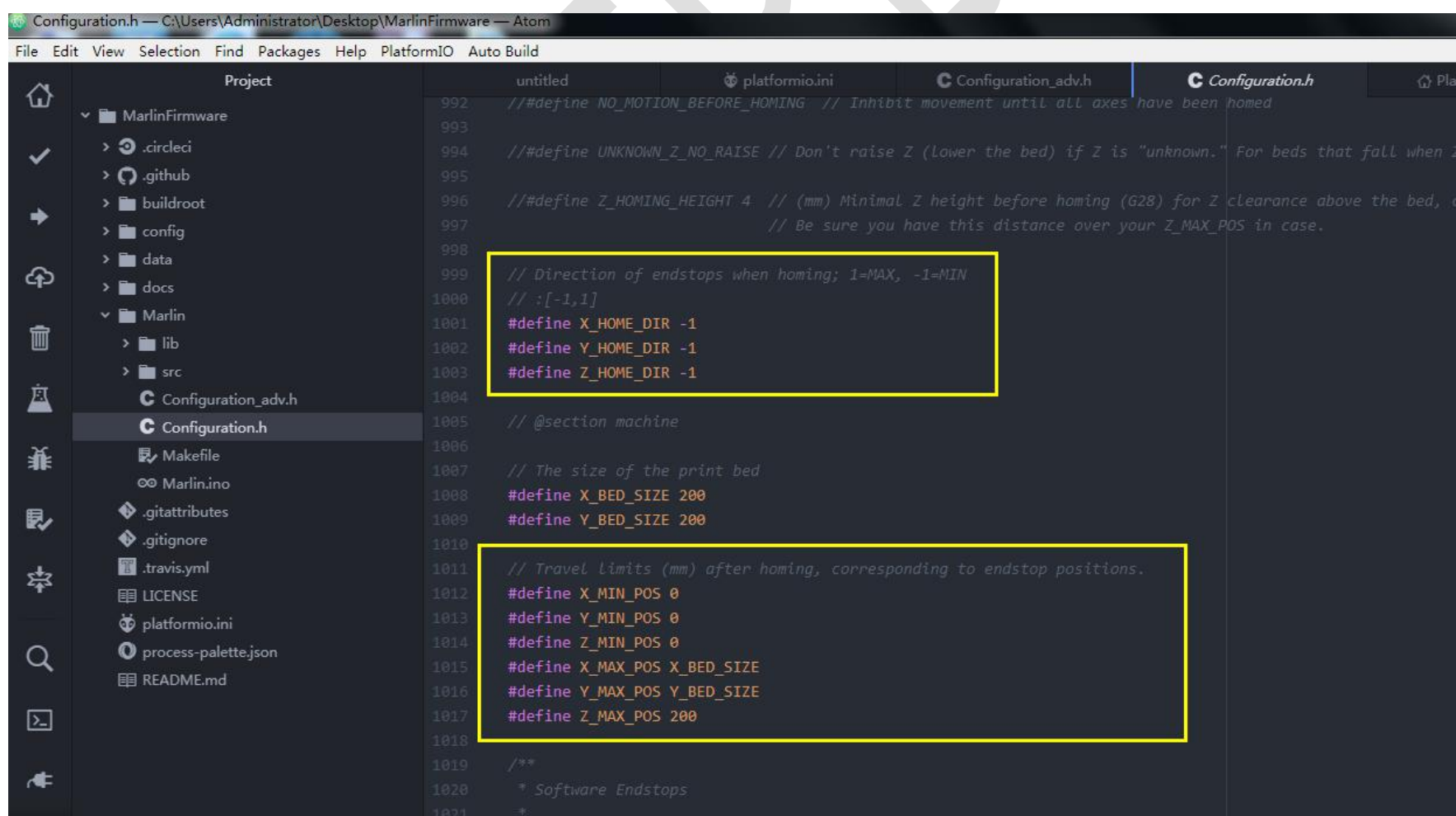
Change the motor direction by “true” and “false”.



```

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 fall when Z is not set.
995
996 // #define Z_HOMING_HEIGHT 4 // (mm) Minimal Z height before homing (G28) for Z clearance above the bed, and
  
```

3) Homing Direction and Travel Limits

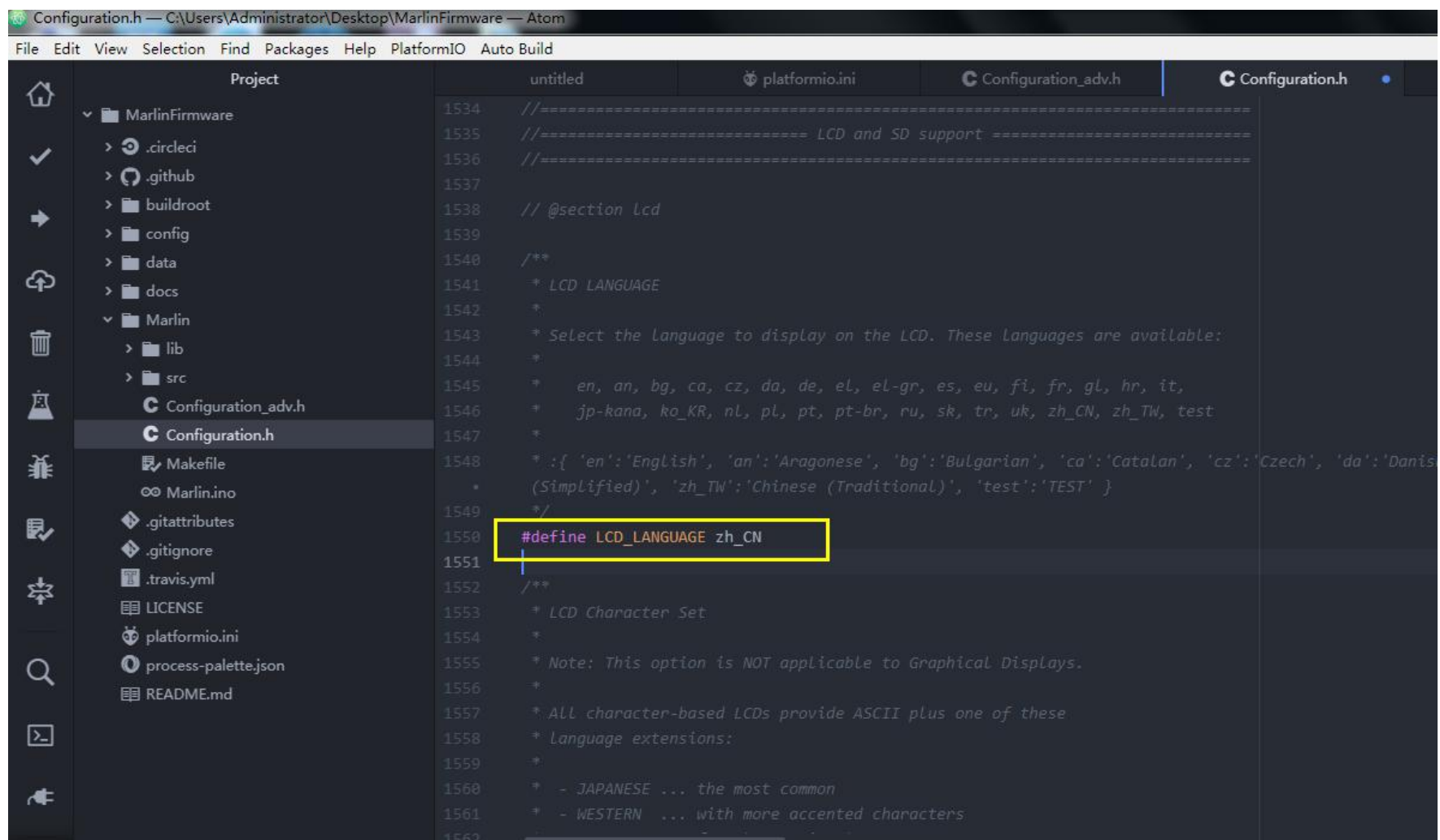


```

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 fall when Z is not set.
995
996 // #define Z_HOMING_HEIGHT 4 // (mm) Minimal Z height before homing (G28) for Z clearance above the bed, and
997 // Be sure you have this distance over your Z_MAX_POS in case.
998
999 // Direction of endstops when homing; 1=MAX, -1=MIN
1000 // :[-1,1]
1001 #define X_HOME_DIR -1
1002 #define Y_HOME_DIR -1
1003 #define Z_HOME_DIR -1
1004
1005 // @section machine
1006
1007 // The size of the print bed
1008 #define X_BED_SIZE 200
1009 #define Y_BED_SIZE 200
1010
1011 // Travel Limits (mm) after homing, corresponding to endstop positions.
1012 #define X_MIN_POS 0
1013 #define Y_MIN_POS 0
1014 #define Z_MIN_POS 0
1015 #define X_MAX_POS X_BED_SIZE
1016 #define Y_MAX_POS Y_BED_SIZE
1017 #define Z_MAX_POS 200
1018
1019 /**
1020 * Software Endstops
1021 *
  
```

8. Section LCD

1) Language Setting



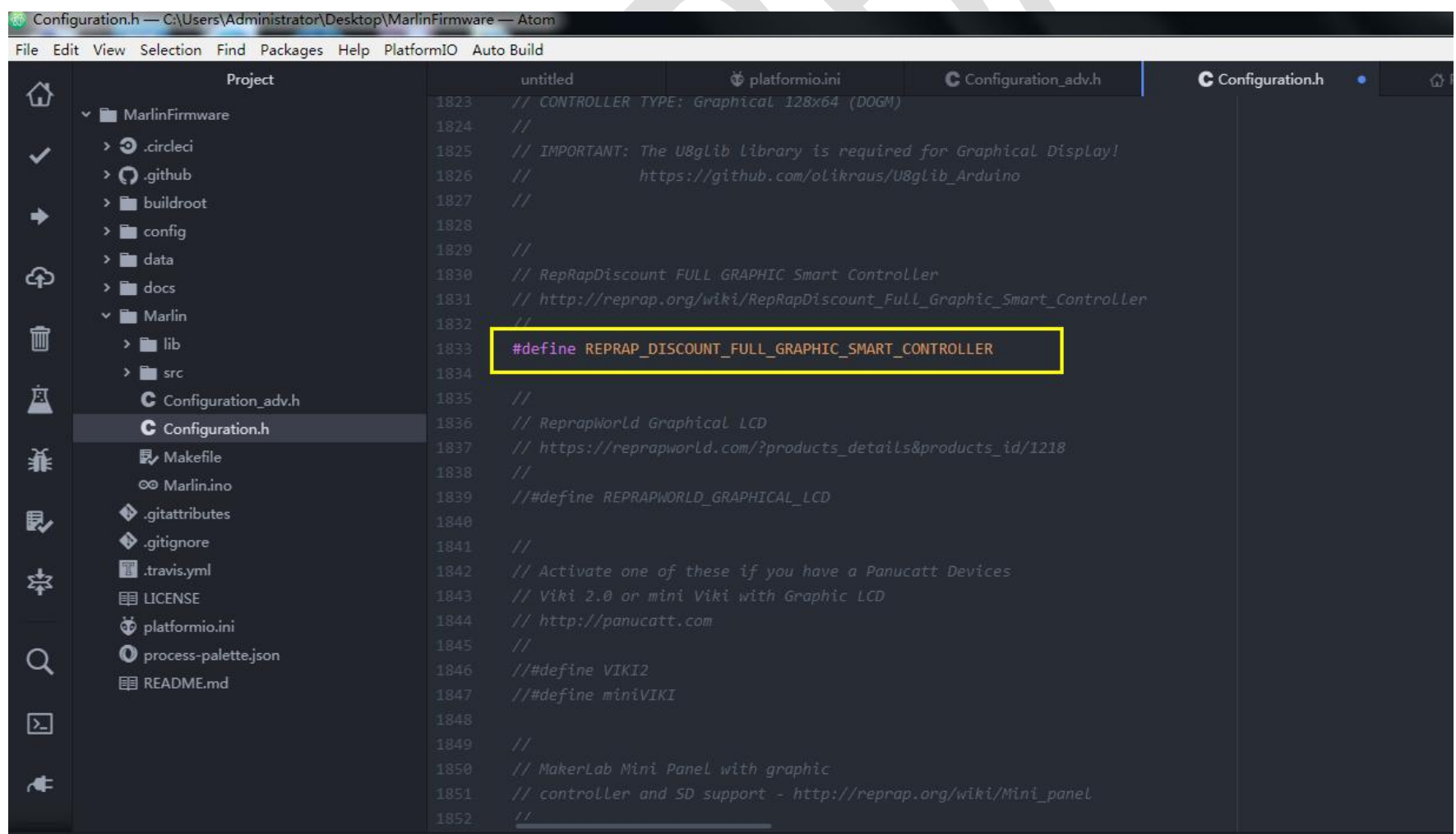
```

1534 //=====
1535 //===== LCD and SD support =====
1536 //=====
1537
1538 // @section lcd
1539
1540 /**
1541  * LCD LANGUAGE
1542  *
1543  * Select the language to display on the LCD. These languages are available:
1544  *
1545  *   en, an, bg, ca, cz, da, de, el, el-gr, es, eu, fi, fr, gl, hr, it,
1546  *   jp-kana, ko_KR, nl, pl, pt, pt-br, ru, sk, tr, uk, zh_CN, zh_TW, test
1547  *
1548  * :{ 'en': 'English', 'an': 'Aragonese', 'bg': 'Bulgarian', 'ca': 'Catalan', 'cz': 'Czech', 'da': 'Danish
1549  *   (Simplified)', 'zh_TW': 'Chinese (Traditional)', 'test': 'TEST' }
1550  */
1551 #define LCD_LANGUAGE zh_CN
1552
1553 /**
1554  * LCD Character Set
1555  *
1556  * Note: This option is NOT applicable to Graphical Displays.
1557  *
1558  * All character-based LCDs provide ASCII plus one of these
1559  * language extensions:
1560  *
1561  * - JAPANESE ... the most common
1562  * - WESTERN ... with more accented characters

```

2) LCD Type Setting

Enable LCD12864

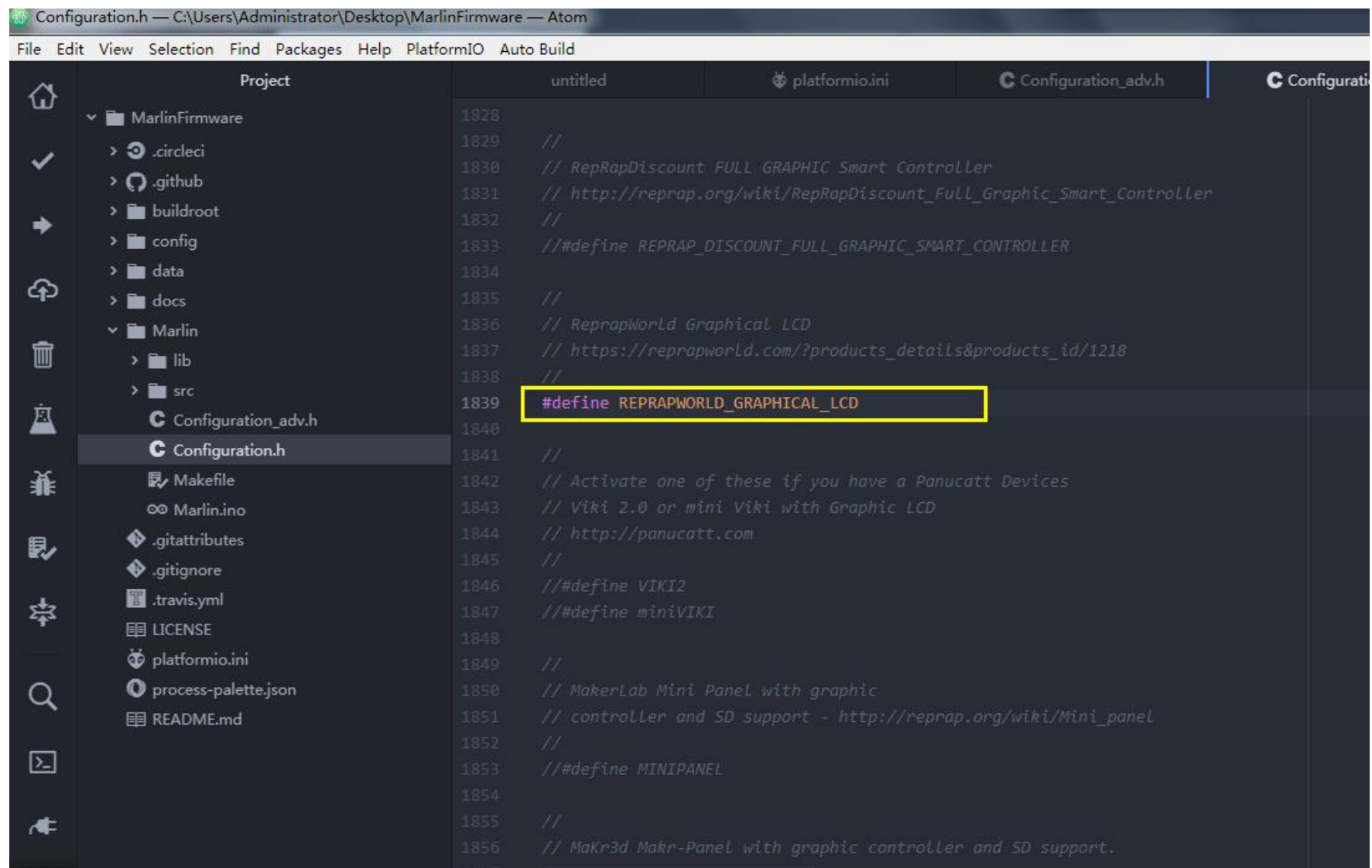


```

1823 // CONTROLLER TYPE: Graphical 128x64 (DOGM)
1824 //
1825 // IMPORTANT: The U8glib library is required for Graphical Display!
1826 //             https://github.com/olikraus/U8glib_Arduino
1827 //
1828
1829 //
1830 // RepRapDiscount FULL GRAPHIC Smart Controller
1831 // http://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller
1832 //
1833 #define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
1834
1835 //
1836 // ReprapWorld Graphical LCD
1837 // https://reprapworld.com/?products_details&products_id/1218
1838 //
1839 // #define REPRAPWORLD_GRAPHICAL_LCD
1840
1841 //
1842 // Activate one of these if you have a Panucatt Devices
1843 // Viki 2.0 or mini Viki with Graphic LCD
1844 // http://panucatt.com
1845 //
1846 // #define VIKI2
1847 // #define miniVIKI
1848
1849 //
1850 // MakerLab Mini Panel with graphic
1851 // controller and SD support - http://reprap.org/wiki/Mini_panel
1852 //

```

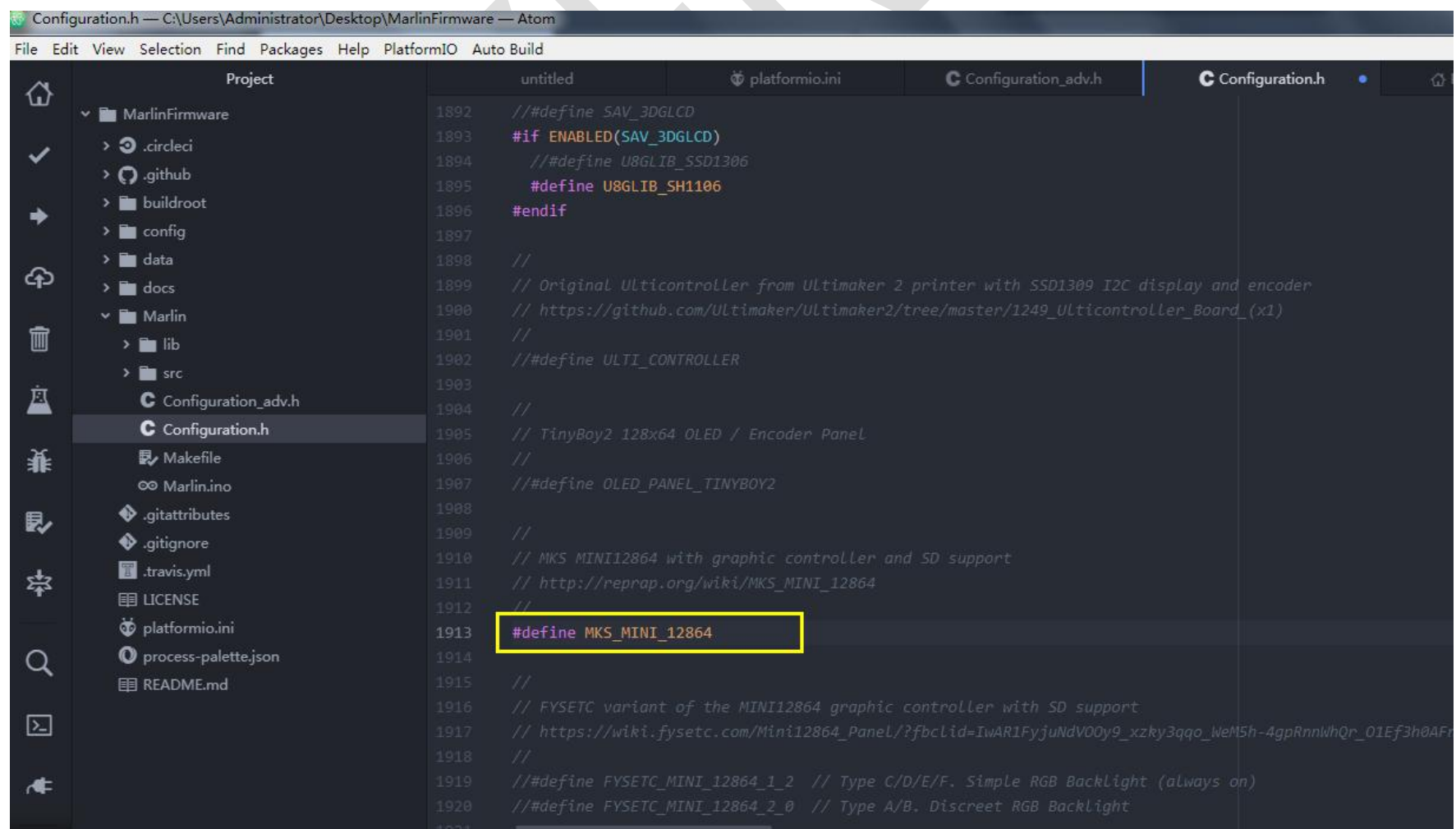
Enable LCD2004



```

1828
1829 //
1830 // RepRapDiscount FULL GRAPHIC Smart Controller
1831 // http://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller
1832 //
1833 // #define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
1834
1835 //
1836 // ReprapWorld Graphical LCD
1837 // https://reprapworld.com/?products_details&products_id/1218
1838 //
1839 #define REPRAPWORLD_GRAPHICAL_LCD
1840
1841 //
1842 // Activate one of these if you have a Panucatt Devices
1843 // Viki 2.0 or mini Viki with Graphic LCD
1844 // http://panucatt.com
1845 //
1846 // #define VIKI2
1847 // #define miniVIKI
1848
1849 //
1850 // MakerLab Mini Panel with graphic
1851 // controller and SD support - http://reprap.org/wiki/Mini_panel
1852 //
1853 // #define MINIPANEL
1854
1855 //
1856 // MaKr3d Makr-Panel with graphic controller and SD support.
1857
  
```

Enable MINI12864 and LCD12864B



```

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-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
  
```