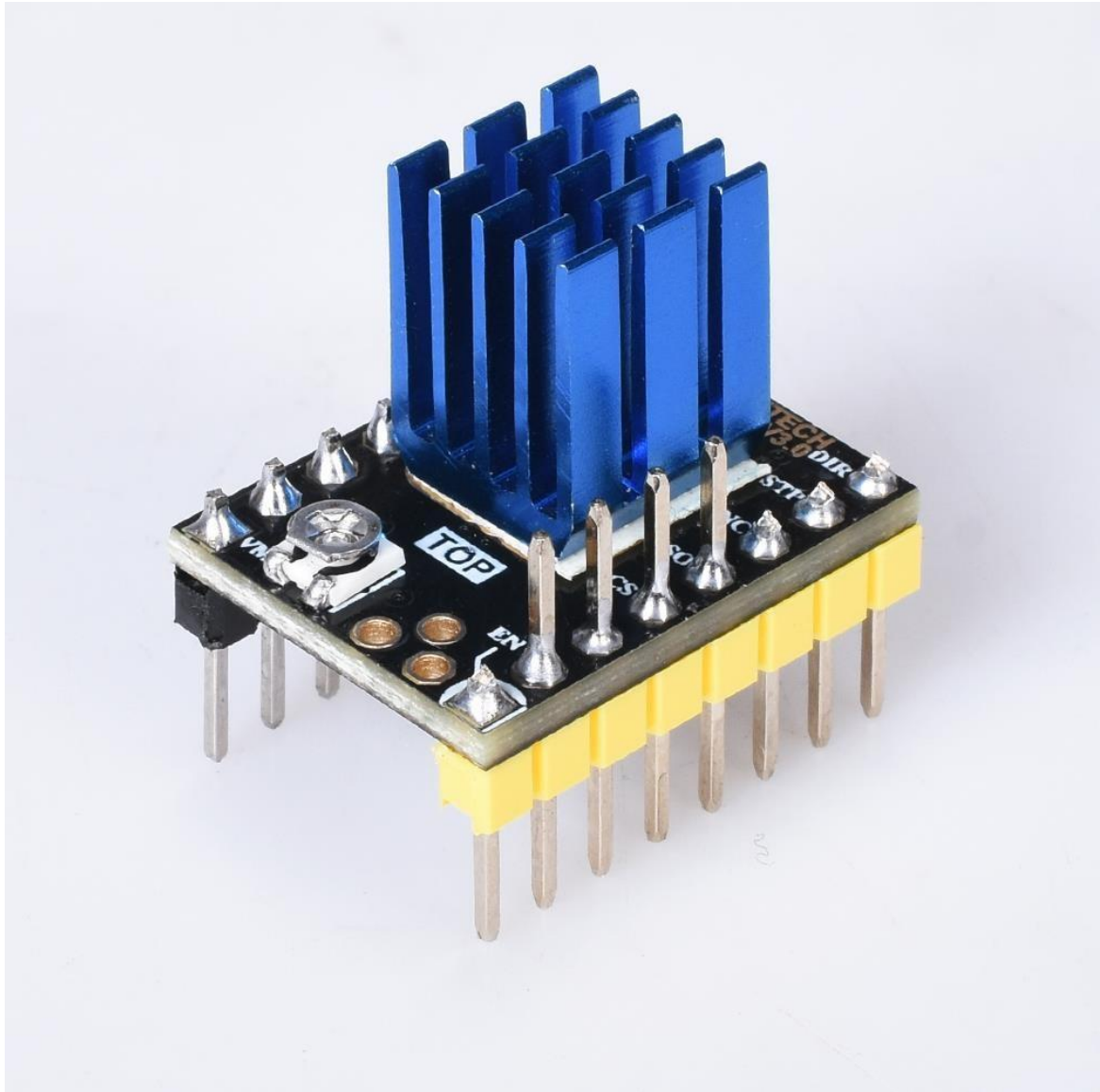


Shen Zhen Big Tree
Technology CO.,LTD.

BIGTREE TECH

TMC2130-V3.0



TMC2130 is a high-performance two-phase stepper motor driver chip with a standard SPI interface and a simple step/dir interface. Very low noise, high load driver does not lose step.

Product features:

1. Up to 256 subdivisions;
2. Support SPI interface control;
3. Do not lose step when driving under high load;
4. Logic voltage: 3.3V/5V;
5. Input voltage range: 12V/24V DC;
6. Motor operating current 1.2a (peak 2A) can be adjusted
7. Spreadcycle ensures that the stepper motor has a smooth transition without dead zone current when passing zero.

TMC2130 stepper motor driver chip has almost no jitter;

8. StealthChop can drive the motor to work under extremely quiet sound, so it can be controlled

The noise of the motor is below 10dB, which is much lower than that of the traditional current controller

Type;

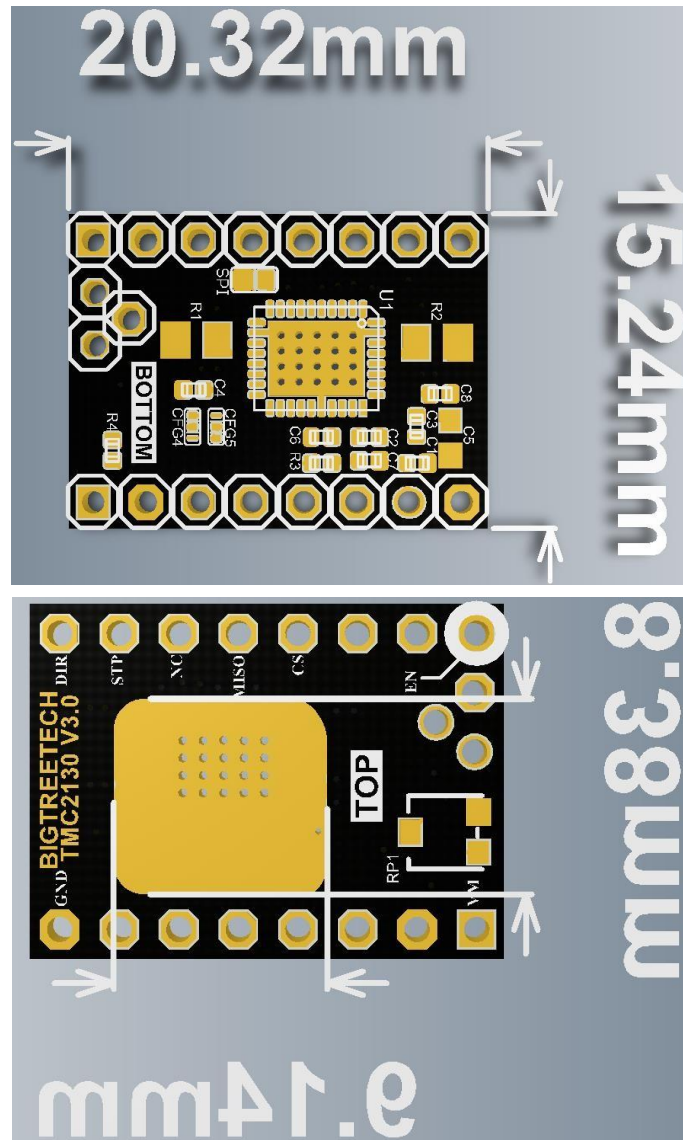
9. SpreadCycle PWM chopped mode is automatically divided between slow and fast attenuators

Setting up a hysteresis attenuation function, the average current reflects the configured normal current in sine

There will be no transition period at the zero crossing of the sine, which reduces the current and torque fluctuations that make the current waveform more similar to the sine wave. Compared with the traditional constant chopper mode of SpreadCycle PWM, the motor operates more smoothly and stably

Shen Zhen Big Tree
Technology CO.,LTD.

Size parameters



Parameter Description:

Highest Resolution 256 microsteps per full step

Step/Dir Interface with microstep interpolation

microPlyer™ SPI Interface

stealthChop™ for extremely quiet operation and smooth motion

spreadCycle™ highly dynamic motor control chopper

coolStep™ current control for energy savings up to 75%

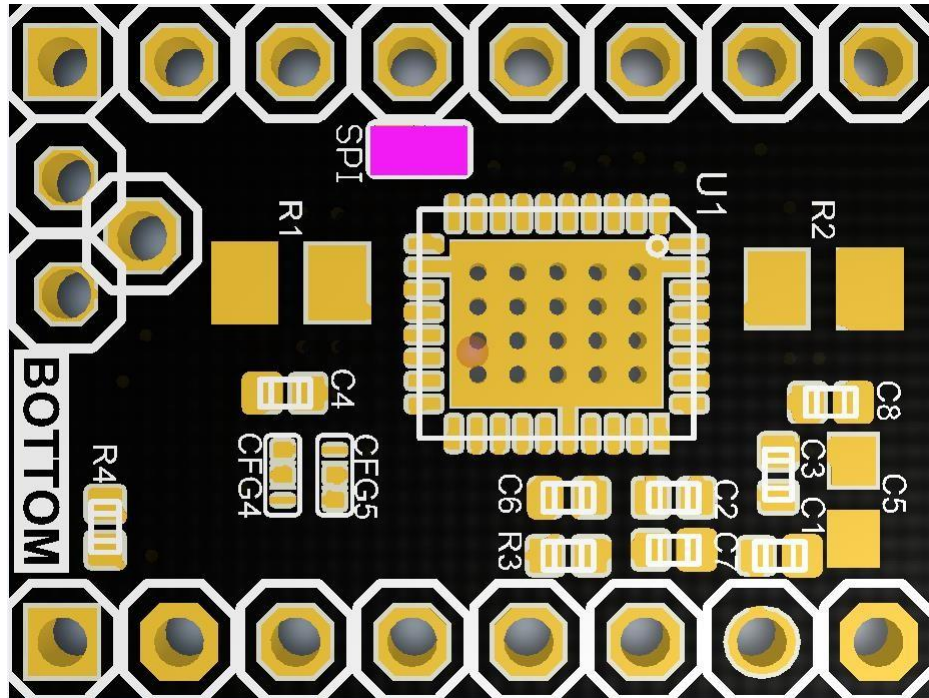
Voltage Range 12V/24VDC

Passive Braking and freewheeling mode

Full Protection & Diagnostics

Working mode :

1.STEP/DIR mode:



(1)SPI position welding 0Ω resistor, making the driver work in STEP/DIR mode.

(2)Working mode selection: SDI (CFG1), SCK (CFG2)

Standalone Operation(SPI_MODE=GND) STP/DIR MODE

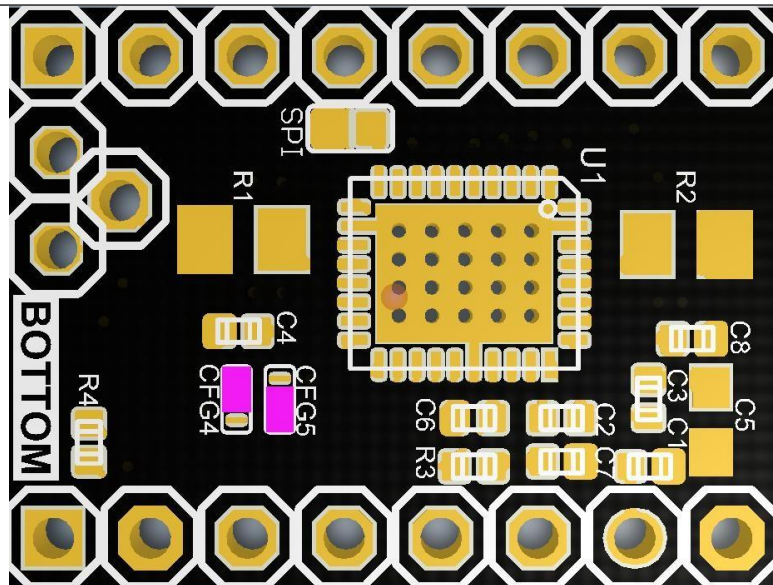
CFG6/EN				
GND -> Driver enable				
Vio -> Driver disable				
Open-> Driver enable with ramp down from 100% to 34% after about 3s				
CFG2 CFG1 Steps Interpolation Chopper Mode				
GND	GND	1	NO	spreadcycle
GND	Vio	2	NO	spreadcycle
GND	Open	2	Yes to 256	spreadcycle
Vio	GND	4	NO	spreadcycle
Vio	Vio	16	NO	spreadcycle
Vio	Open	4	Yes to 256	spreadcycle
Open	GND	16	Yes to 256	spreadcycle
Open	Vio	4	Yes to 256	stealthchop
Open	Open	16	Yes to 256	stealthchop

2. SPI working mode wiring instructions:

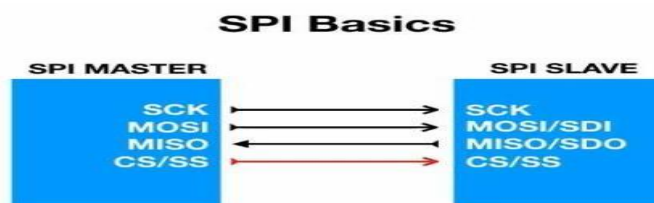
Before wiring, it is necessary to select the hardware operation mode of the driver module:

(1)1. Solder **CFG4** and **CFG5** as shown in the purple area (CFG4 is connected to GND, CFG5 is connected to VCC); (Welding is correct to enable spreadcycle mode)

(2)Remove the resistor at the SPI position to put the driver in SPI mode



Wiring diagram is as follows:



Three lines common to all the devices:

MOSI (Master Out Slave In)

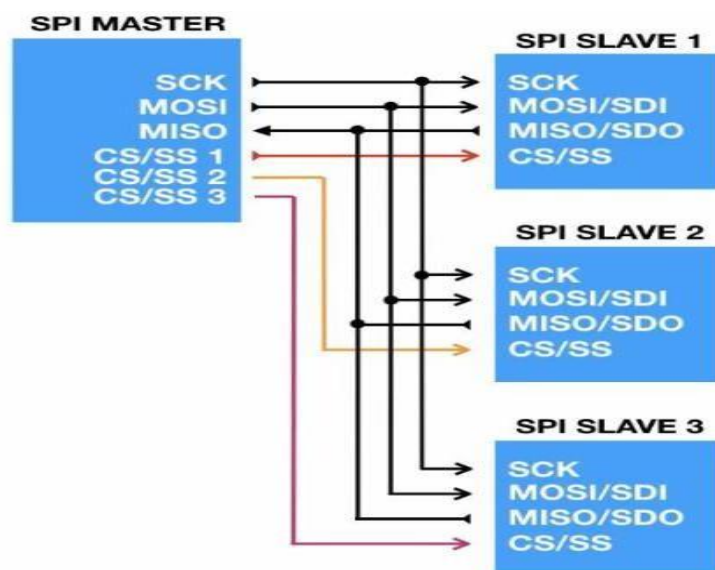
MISO (Master In Slave Out)

SCK (Serial Clock)

One line specific for every device:

SS (Slave Select) / CS (Chip Select)

Wiring diagram:



3. Potentiometer adjustment instructions

Rotate the potentiometer clockwise: reduce Vref to reduce the driver current;

Turn the potentiometer counterclockwise: Increase Vref to increase the driver current.

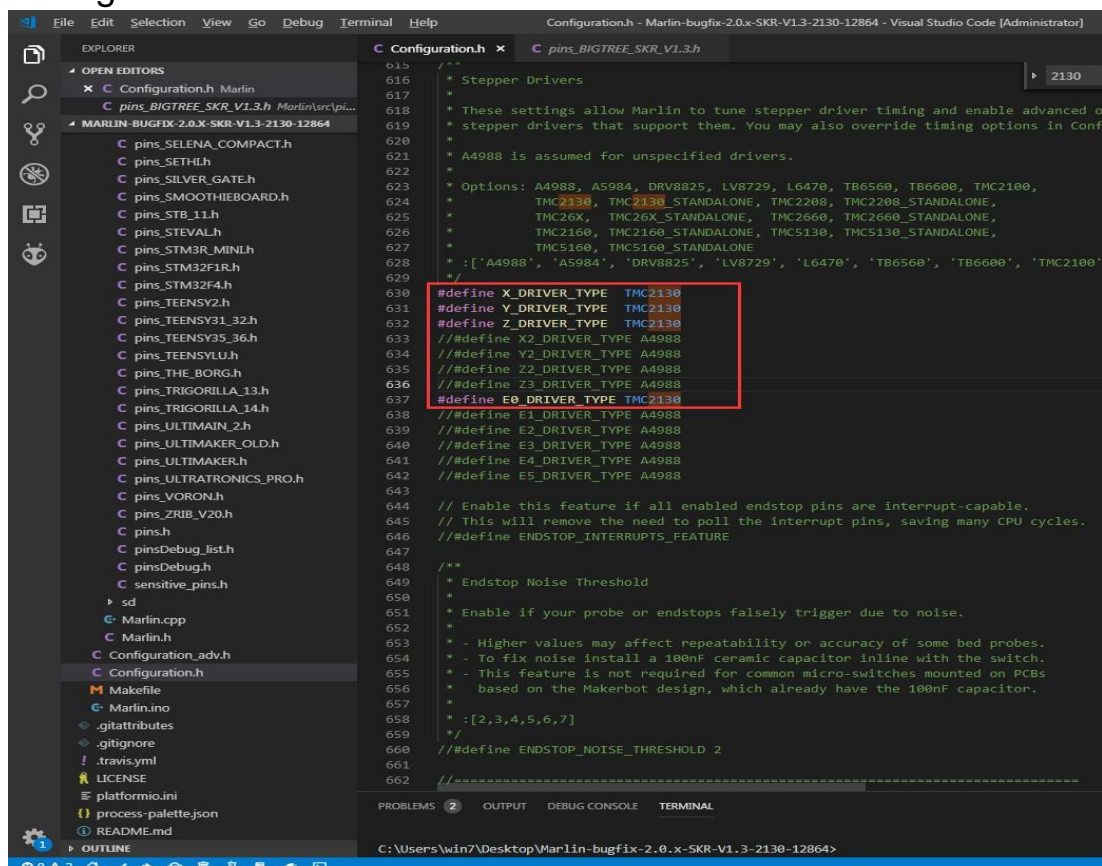
The exact voltage of Vref must be measured when the main board is supplied with 12V or 24V.

Range of Vref values: default: 1V (± 0.2); MAX: 2.4V; MIN: 0V;

Do not use too much force when rotating the potentiometer to prevent irreversible damage to the potentiometer; when it is rotated counterclockwise to the maximum value, it will become the minimum value if it continues to rotate; similarly, when it is rotated clockwise to the minimum value, If you continue to rotate, it will become the maximum value;

SPI mode firmware change instructions:

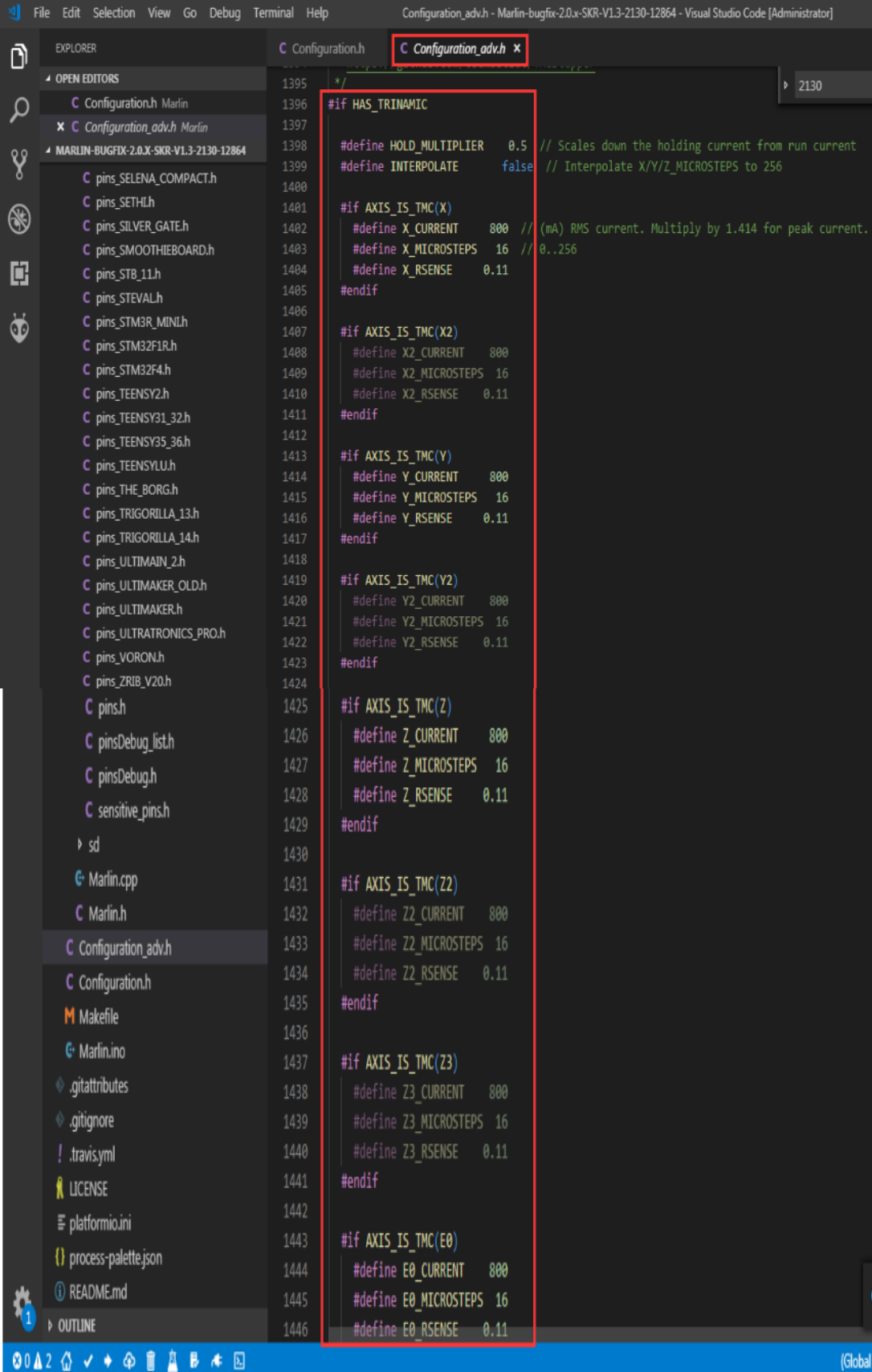
For example, (Marlin-2.0)BIGTREETECH SKR V1.3 :
Configuration.h file:



```
615  /**
616  * Stepper Drivers
617  *
618  * These settings allow Marlin to tune stepper driver timing and enable advanced o
619  * stepper drivers that support them. You may also override timing options in Conf
620  *
621  * A4988 is assumed for unspecified drivers.
622  *
623  * Options: A4988, A5984, DRV8825, LV8729, L6470, TB6600, TB6600, TMC2100,
624  *           TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
625  *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
626  *           TMC2160, TMC2160_STANDALONE, TMC5130, TMC5130_STANDALONE,
627  *           TMC5160, TMC5160_STANDALONE
628  * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'TB6600', 'TB6600', 'TMC2100'
629  */
630
631 #define X_DRIVER_TYPE  TMC2130
632 #define Y_DRIVER_TYPE  TMC2130
633 #define Z_DRIVER_TYPE  TMC2130
634 // #define X2_DRIVER_TYPE A4988
635 // #define Y2_DRIVER_TYPE A4988
636 // #define Z2_DRIVER_TYPE A4988
637 #define E0_DRIVER_TYPE TMC2130
638 // #define E1_DRIVER_TYPE A4988
639 // #define E2_DRIVER_TYPE A4988
640 // #define E3_DRIVER_TYPE A4988
641 // #define E4_DRIVER_TYPE A4988
642 // #define E5_DRIVER_TYPE A4988
643
644 // Enable this feature if all enabled endstop pins are interrupt-capable.
645 // This will remove the need to poll the interrupt pins, saving many CPU cycles.
646 // #define ENDSTOP_INTERRUPTS_FEATURE
647
648 /**
649  * Endstop Noise Threshold
650  *
651  * Enable if your probe or endstops falsely trigger due to noise.
652  *
653  * - Higher values may affect repeatability or accuracy of some bed probes.
654  * - To fix noise install a 100nF ceramic capacitor inline with the switch.
655  * - This feature is not required for common micro-switches mounted on PCBs
656  *   based on the Makerbot design, which already have the 100nF capacitor.
657  *
658  * : [2,3,4,5,6,7]
659  */
660 // #define ENDSTOP_NOISE_THRESHOLD 2
661
662 //=====
```

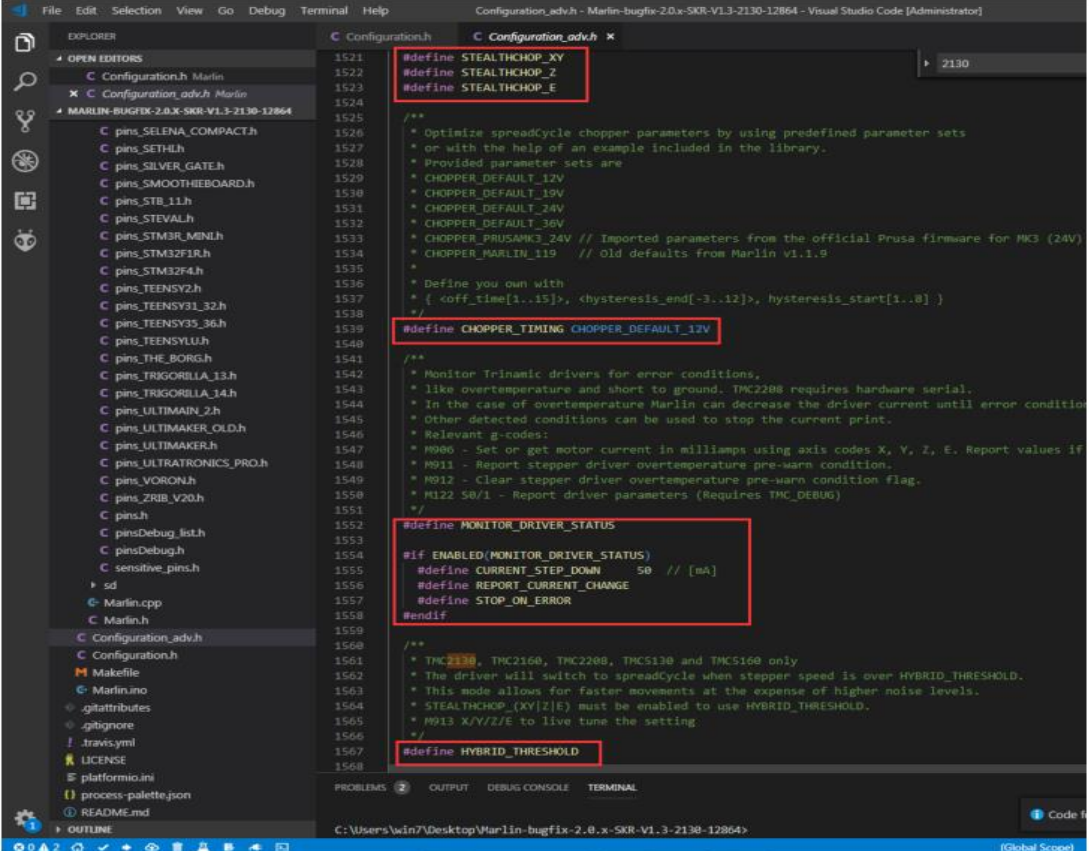
Shen Zhen Big Tree Technology CO.,LTD.

Configuration_adv.h file:

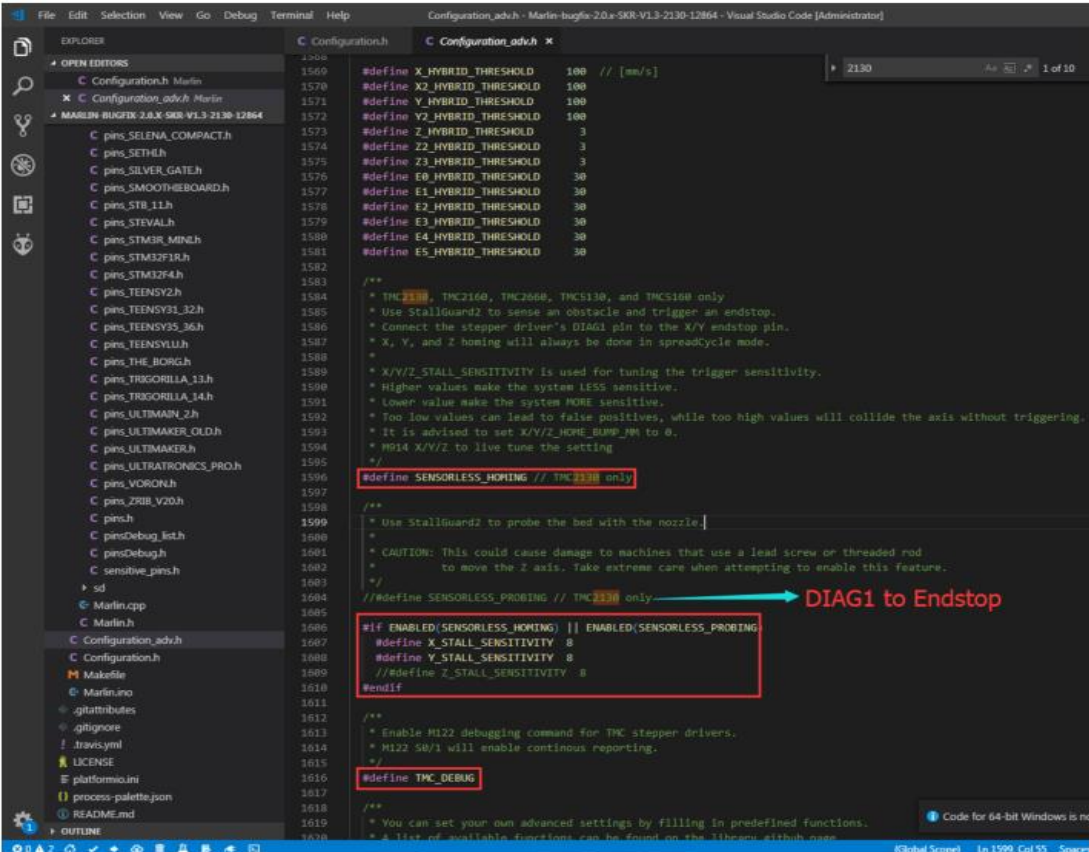


```
1395 */
1396 #if HAS_TRINAMIC
1397
1398 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
1399 #define INTERPOLATE false // Interpolate X/Y/Z_MICROSTEPS to 256
1400
1401 #if AXIS_IS_TMC(X)
1402 #define X_CURRENT 800 // (mA) RMS current. Multiply by 1.414 for peak current.
1403 #define X_MICROSTEPS 16 // 0..256
1404 #define X_RSENSE 0.11
1405 #endif
1406
1407 #if AXIS_IS_TMC(X2)
1408 #define X2_CURRENT 800
1409 #define X2_MICROSTEPS 16
1410 #define X2_RSENSE 0.11
1411 #endif
1412
1413 #if AXIS_IS_TMC(Y)
1414 #define Y_CURRENT 800
1415 #define Y_MICROSTEPS 16
1416 #define Y_RSENSE 0.11
1417 #endif
1418
1419 #if AXIS_IS_TMC(Y2)
1420 #define Y2_CURRENT 800
1421 #define Y2_MICROSTEPS 16
1422 #define Y2_RSENSE 0.11
1423 #endif
1424
1425 #if AXIS_IS_TMC(Z)
1426 #define Z_CURRENT 800
1427 #define Z_MICROSTEPS 16
1428 #define Z_RSENSE 0.11
1429 #endif
1430
1431 #if AXIS_IS_TMC(Z2)
1432 #define Z2_CURRENT 800
1433 #define Z2_MICROSTEPS 16
1434 #define Z2_RSENSE 0.11
1435 #endif
1436
1437 #if AXIS_IS_TMC(Z3)
1438 #define Z3_CURRENT 800
1439 #define Z3_MICROSTEPS 16
1440 #define Z3_RSENSE 0.11
1441 #endif
1442
1443 #if AXIS_IS_TMC(E0)
1444 #define E0_CURRENT 800
1445 #define E0_MICROSTEPS 16
1446 #define E0_RSENSE 0.11
```

Shen Zhen Big Tree Technology CO.,LTD.



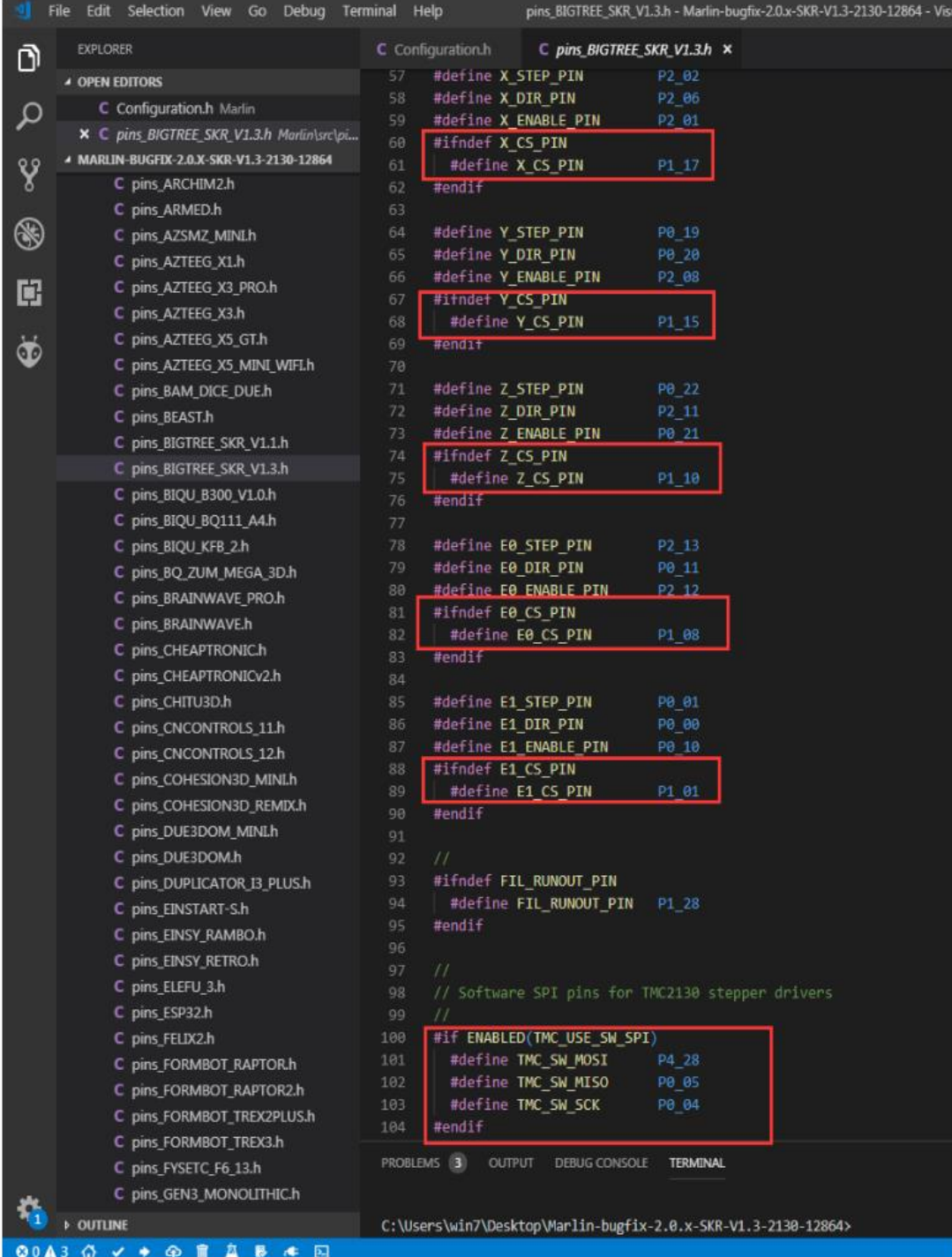
```
1521 #define STEALTHCHOP_XY
1522 #define STEALTHCHOP_Z
1523 #define STEALTHCHOP_E
1524
1525 /**
1526  * Optimize spreadCycle chopper parameters by using predefined parameter sets
1527  * or with the help of an example included in the library.
1528  * Provided parameter sets are
1529  * CHOPPER_DEFAULT_12V
1530  * CHOPPER_DEFAULT_10V
1531  * CHOPPER_DEFAULT_24V
1532  * CHOPPER_DEFAULT_36V
1533  * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
1534  * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
1535  *
1536  * Define your own with
1537  * <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8]
1538  */
1539 #define CHOPPER_TIMING CHOPPER_DEFAULT_12V
1540
1541 /**
1542  * Monitor Trinamic drivers for error conditions.
1543  * like overtemperature and short to ground. TMC2208 requires hardware serial.
1544  * In the case of overtemperature Marlin can decrease the driver current until error condition
1545  * Other detected conditions can be used to stop the current print.
1546  * Relevant g-codes:
1547  * M900 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if
1548  * M911 - Report stepper driver overtemperature pre-warn condition.
1549  * M912 - Clear stepper driver overtemperature pre-warn condition flag.
1550  * M122 S0/1 - Report driver parameters (Requires TMC_DEBUG)
1551  */
1552 #define MONITOR_DRIVER_STATUS
1553
1554 #if ENABLED(MONITOR_DRIVER_STATUS)
1555 #define CURRENT_STEP_DOWN 50 // [mA]
1556 #define REPORT_CURRENT_CHANGE
1557 #define STOP_ON_ERROR
1558 #endif
1559
1560 /**
1561  * TMC2130, TMC2160, TMC2208, TMC5130 and TMC5160 only
1562  * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
1563  * This mode allows for faster movements at the expense of higher noise levels.
1564  * STEALTHCHOP(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
1565  * M913 X/Y/Z/E to live tune the setting
1566  */
1567 #define HYBRID_THRESHOLD
1568
```



```
1569 #define X_HYBRID_THRESHOLD 100 // [mm/s]
1570 #define X2_HYBRID_THRESHOLD 100
1571 #define Y_HYBRID_THRESHOLD 100
1572 #define Y2_HYBRID_THRESHOLD 100
1573 #define Z_HYBRID_THRESHOLD 3
1574 #define Z2_HYBRID_THRESHOLD 3
1575 #define E0_HYBRID_THRESHOLD 30
1576 #define E1_HYBRID_THRESHOLD 30
1577 #define E2_HYBRID_THRESHOLD 30
1578 #define E3_HYBRID_THRESHOLD 30
1579 #define E4_HYBRID_THRESHOLD 30
1580 #define E5_HYBRID_THRESHOLD 30
1581
1582 /**
1583  * TMC2130, TMC2160, TMC2208, TMC5130, and TMC5160 only
1584  * Use StallGuard2 to sense an obstacle and trigger an endstop.
1585  * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
1586  * X, Y, and Z homing will always be done in spreadCycle mode.
1587  *
1588  * X/Y/Z_STALL_SENSITIVITY is used for tuning the trigger sensitivity.
1589  * Higher values make the system LESS sensitive.
1590  * Lower values make the system MORE sensitive.
1591  * Too low values can lead to false positives, while too high values will collide the axis without triggering.
1592  * It is advised to set X/Y/Z_HOMING_BUMP to 0.
1593  * M914 X/Y/Z to live tune the setting
1594  */
1595 #define SENSORLESS_HOMING // TMC2130 only
1596
1597 /**
1598  * Use StallGuard2 to probe the bed with the nozzle.
1599  *
1600  * CAUTION: This could cause damage to machines that use a lead screw or threaded rod
1601  * to move the Z axis. Take extreme care when attempting to enable this feature.
1602  */
1603 // #define SENSORLESS_PROBING // TMC2130 only
1604
1605 #if ENABLED(SENSORLESS_HOMING) || ENABLED(SENSORLESS_PROBING)
1606 #define X_STALL_SENSITIVITY 8
1607 #define Y_STALL_SENSITIVITY 8
1608 // #define Z_STALL_SENSITIVITY 8
1609 #endif
1610
1611 /**
1612  * Enable M122 debugging command for TMC stepper drivers.
1613  * M122 S0/1 will enable continuous reporting.
1614  */
1615 #define TMC_DEBUG
1616
1617 /**
1618  * You can set your own advanced settings by filling in predefined functions.
1619  * A list of available functions can be found on the library github page.
1620  */
1621
```


Shen Zhen Big Tree Technology CO.,LTD.

pins_BIGTREE_SKR_V1.3.h file:



```
57 #define X_STEP_PIN      P2_02
58 #define X_DIR_PIN       P2_06
59 #define X_ENABLE_PIN     P2_01
60 #ifndef X_CS_PIN
61   #define X_CS_PIN       P1_17
62 #endif
63
64 #define Y_STEP_PIN      P0_19
65 #define Y_DIR_PIN       P0_20
66 #define Y_ENABLE_PIN    P2_08
67 #ifndef Y_CS_PIN
68   #define Y_CS_PIN       P1_15
69 #endif
70
71 #define Z_STEP_PIN      P0_22
72 #define Z_DIR_PIN       P2_11
73 #define Z_ENABLE_PIN    P0_21
74 #ifndef Z_CS_PIN
75   #define Z_CS_PIN       P1_10
76 #endif
77
78 #define E0_STEP_PIN     P2_13
79 #define E0_DIR_PIN      P0_11
80 #define E0_ENABLE_PIN   P2_12
81 #ifndef E0_CS_PIN
82   #define E0_CS_PIN      P1_08
83 #endif
84
85 #define E1_STEP_PIN     P0_01
86 #define E1_DIR_PIN      P0_00
87 #define E1_ENABLE_PIN   P0_10
88 #ifndef E1_CS_PIN
89   #define E1_CS_PIN      P1_01
90 #endif
91
92 //
93 #ifndef FIL_RUNOUT_PIN
94   #define FIL_RUNOUT_PIN P1_28
95 #endif
96
97 //
98 // Software SPI pins for TMC2130 stepper drivers
99 //
100 #if ENABLED(TMC_USE_SW_SPI)
101   #define TMC_SW_MOSI    P4_28
102   #define TMC_SW_MISO    P0_05
103   #define TMC_SW_SCK     P0_04
104 #endif
```

After firmware debugging, compile and upload to the motherboard, open [epronterface](#) online Print software, connected to the motherboard can view the SPI mode 2130 drive Dynamic running state.

**Shen Zhen Big Tree
Technology CO.,LTD.**

Note:

- (1) When selecting SPI operation mode for hardware, please use soldering iron carefully to avoid burns. After processing, observe carefully whether there is residual tin residue in the module, and clean it up to prevent short-circuit and burning of the module;
- (2) Please pay attention to the wire sequence and IO port when wiring. If the wrong wire is connected, the drive will not work. Connect them as shown above carefully .
- (3) When the driver is inserted into the motherboard, please pay attention to see the direction of the drive, can not be inserted in reverse, to prevent the drive from burning;
- (4) Be sure to do a good job before driving heat dissipation work (heat sink + fan), to prevent the drive is not working properly

If you have problems in use, welcome to contact us, we will be careful to answer for you;

If you have any good comments or suggestions on our products, please feel free to give us your feedback.

We will also carefully consider your comments or Suggestions, thank you for choosing BIGTREETECH Product, thank you!