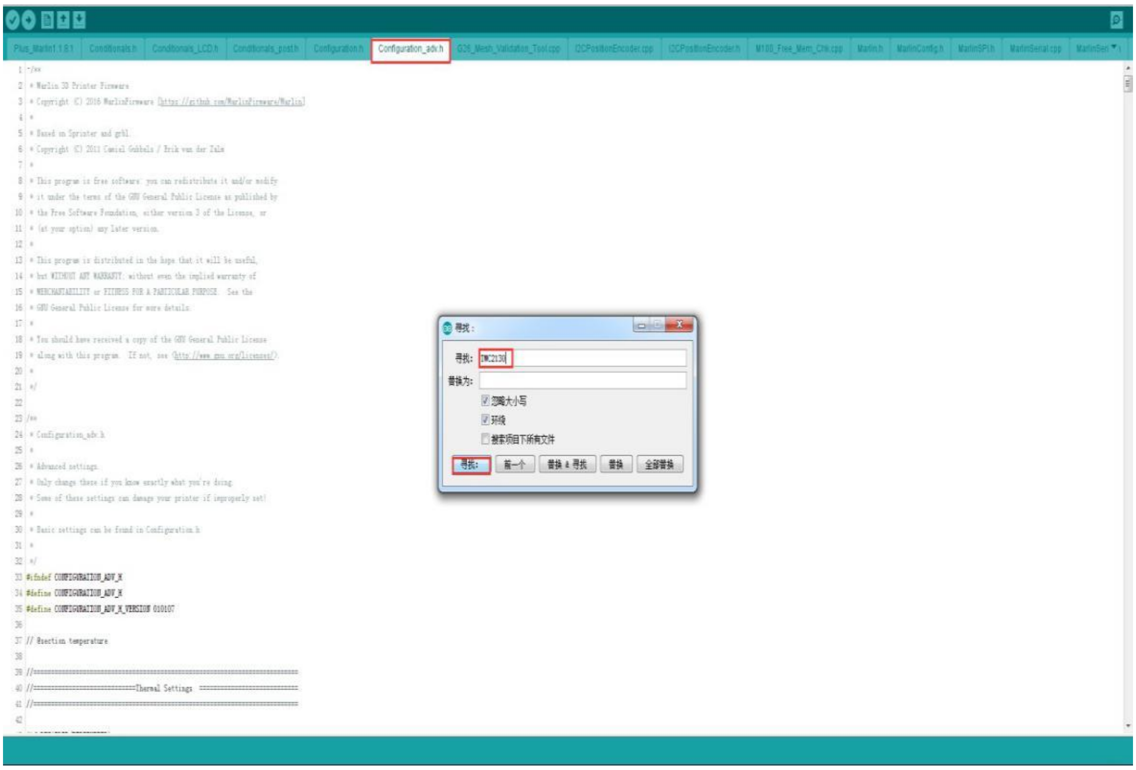
ROBOTER
Bausatz-Shop

TMC2130 Firmware change instructions

1. Open the firmware you want to use using the latest version of Arduino. Find configuration_adv.h.

Then search TMC2130:



```
/**
 * Marlin 3D Printer Firmware
 * Copyright (C) 2016 MarlinFirmware [https://github.com/MarlinFirmware/Marlin]
 *
 * Based on Sprinter and grbl.
 * Copyright (C) 2011 Camiel Gubbels / Erik van der Zalm
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 */

/**
 * Configuration_adv.h
 *
 * Advanced settings.
 * Only change these if you know exactly what you're doing.
 * Some of these settings can damage your printer if improperly set!
 *
 * Basic settings can be found in Configuration.h
 *
 */
#ifndef CONFIGURATION_ADV_H
#define CONFIGURATION_ADV_H
#define CONFIGURATION_ADV_H_VERSION 010107

// @section temperature

//===========================================================================
//=============================Thermal Settings  ============================
//===========================================================================
```

寻找: TMC2130

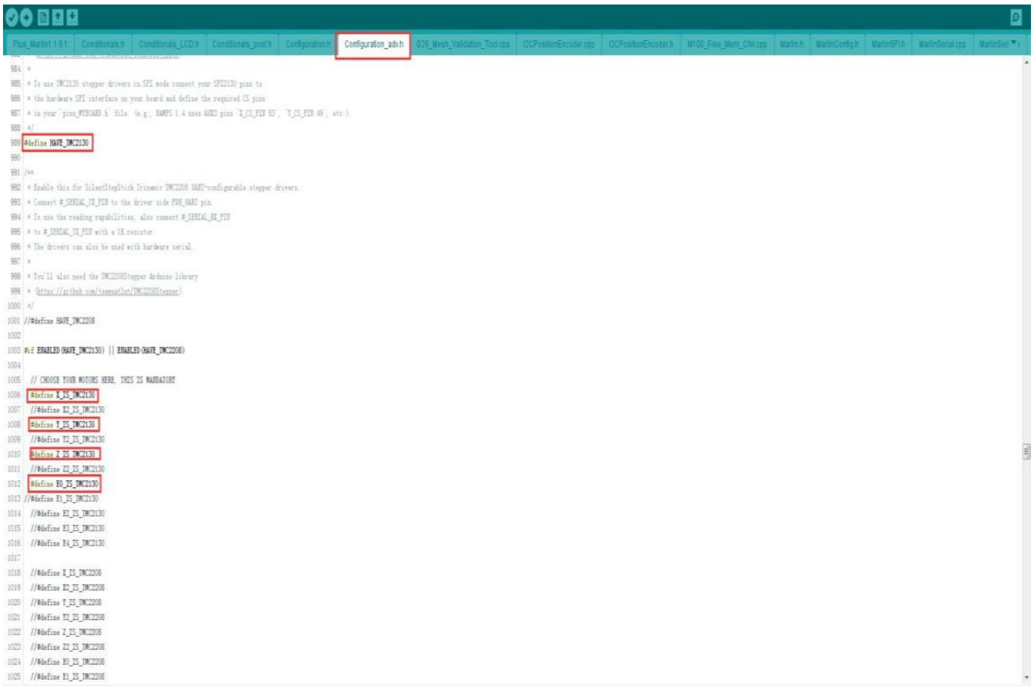
☑ 忽略大小写

☑ 环绕

□ 搜索项目下所有文件

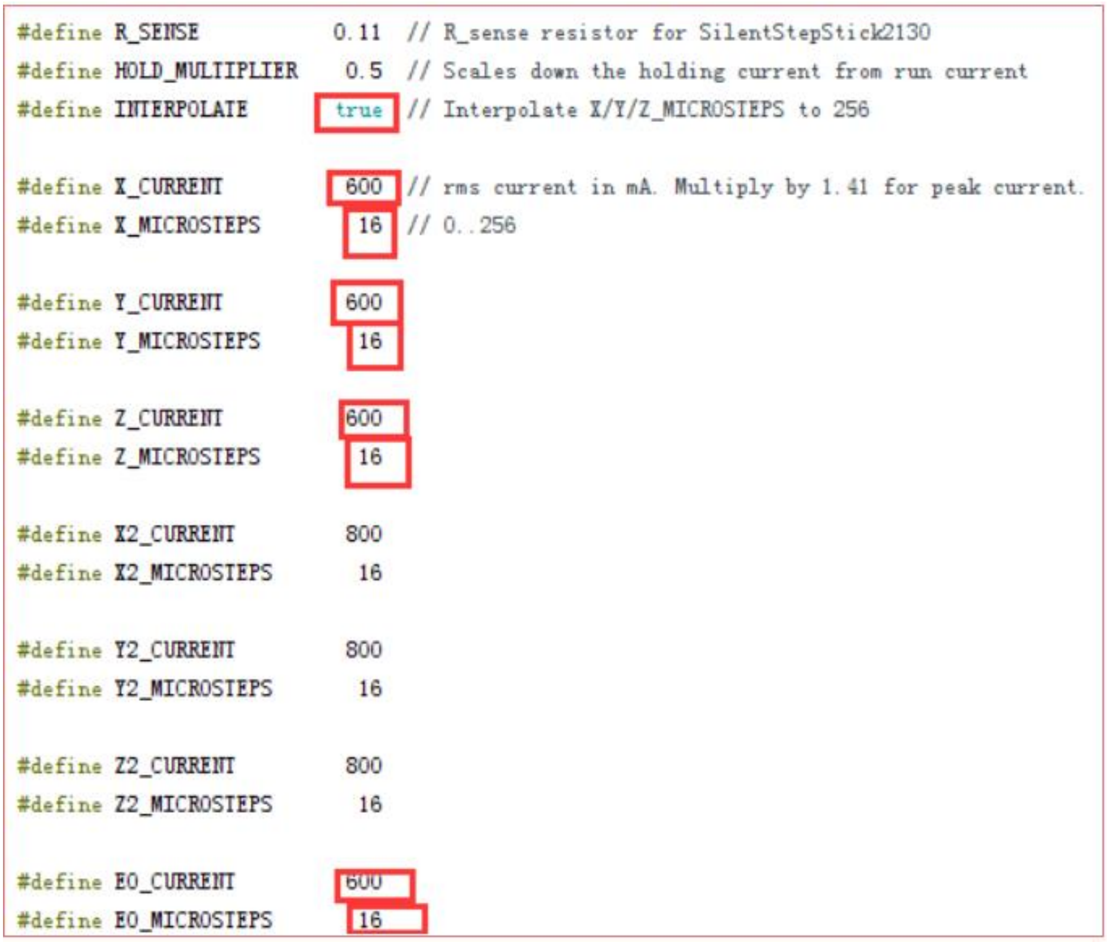
2 find // #define HAVE_TMC2130, and then delete the shield character "//" in the front, to the following // #define X_IS_TMC2130, // #define Y_IS_TMC2130, // #define

Z_IS_TMC2130, // #define E0_IS_TMC2130



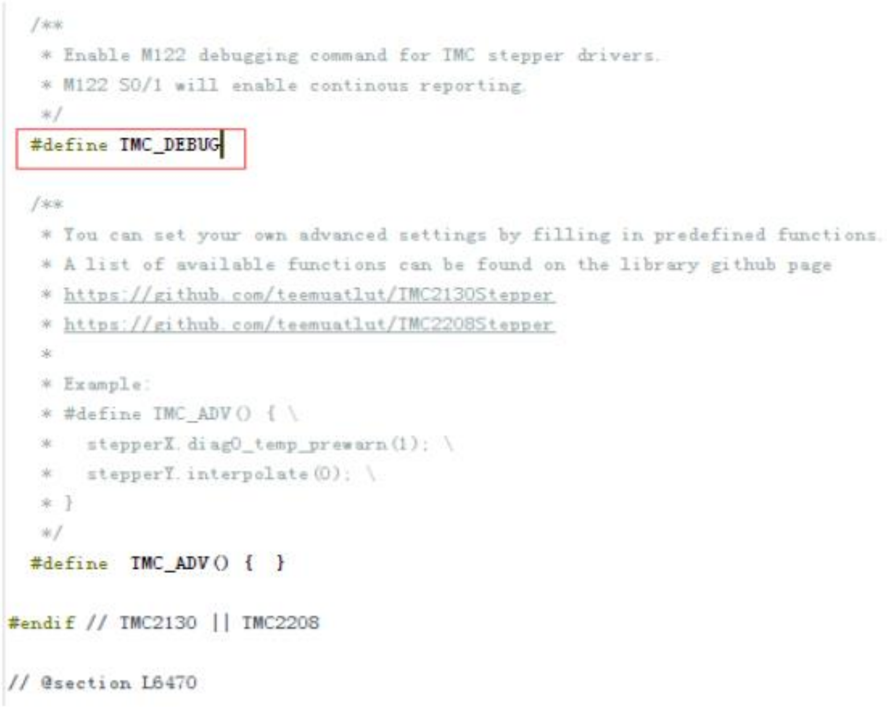
```
 *
 * To use TMC2130 stepper drivers in SPI mode connect your SPI2130 pins to
 * the hardware SPI interface on your board and define the required CS pins
 * in your `pins_MYBOARD.h` file. (e.g., RAMPS 1.4 uses AUX3 pins `X_CS_PIN 53`, `Y_CS_PIN 49`, etc.).
 */
#define HAVE_TMC2130

/**
 * Enable this for SilentStepStick Trinamic TMC2208 UART-configurable stepper drivers.
 * Connect #_SERIAL_TX_PIN to the driver side PDN_UART pin.
 * To use the reading capabilities, also connect #_SERIAL_RX_PIN
 * to #_SERIAL_TX_PIN with a 1K resistor.
 * The drivers can also be used with hardware serial.
 *
 * You'll also need the TMC2208Stepper Arduino library
 * (https://github.com/teemuatlut/TMC2208Stepper)
 */
//#define HAVE_TMC2208

#if ENABLED(HAVE_TMC2130) || ENABLED(HAVE_TMC2208)

  // CHOOSE YOUR MOTORS HERE, THIS IS MANDATORY
  #define X_IS_TMC2130
  //#define X2_IS_TMC2130
  #define Y_IS_TMC2130
  //#define Y2_IS_TMC2130
  #define Z_IS_TMC2130
  //#define Z2_IS_TMC2130
  #define E0_IS_TMC2130
//#define E1_IS_TMC2130
  //#define E2_IS_TMC2130
  //#define E3_IS_TMC2130
  //#define E4_IS_TMC2130

  //#define X_IS_TMC2208
  //#define X2_IS_TMC2208
  //#define Y_IS_TMC2208
  //#define Y2_IS_TMC2208
  //#define Z_IS_TMC2208
  //#define Z2_IS_TMC2208
  //#define E0_IS_TMC2208
  //#define E1_IS_TMC2208
```

3 then change the driving current and subdivision of each driver (select values according to the motor you use)



```
#define R_SENSE           0.11  // R_sense resistor for SilentStepStick2130
#define HOLD_MULTIPLIER    0.5  // Scales down the holding current from run current
#define INTERPOLATE       true  // Interpolate X/Y/Z_MICROSTEPS to 256

#define X_CURRENT          600  // rms current in mA. Multiply by 1.41 for peak current.
#define X_MICROSTEPS        16  // 0..256

#define Y_CURRENT          600
#define Y_MICROSTEPS        16

#define Z_CURRENT          600
#define Z_MICROSTEPS        16

#define X2_CURRENT         800
#define X2_MICROSTEPS       16

#define Y2_CURRENT         800
#define Y2_MICROSTEPS       16

#define Z2_CURRENT         800
#define Z2_MICROSTEPS       16

#define E0_CURRENT         600
#define E0_MICROSTEPS       16
```

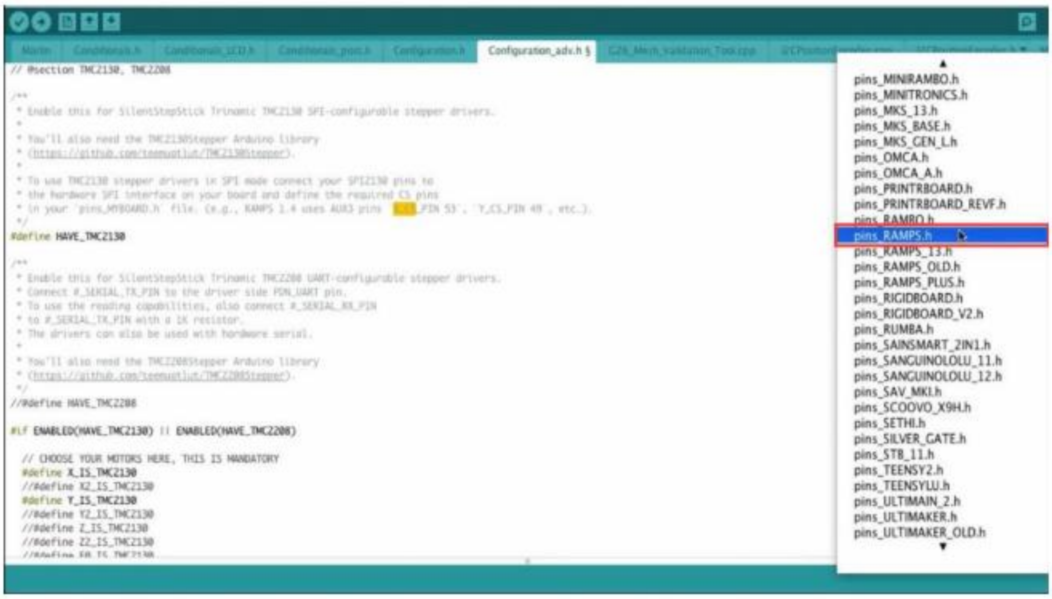
4. Unmask the // #define TMC_DEBUG statement for later detection:



```
  /**
   * Enable M122 debugging command for TMC stepper drivers.
   * M122 S0/1 will enable continous reporting.
   */
  #define TMC_DEBUG

  /**
   * You can set your own advanced settings by filling in predefined functions.
   * A list of available functions can be found on the library github page
   * https://github.com/teemuatlut/TMC2130Stepper
   * https://github.com/teemuatlut/TMC2208Stepper
   *
   * Example:
   * #define TMC_ADV() { \
   *   stepperX.diag0_temp_prewarn(1); \
   *   stepperY.interpolate(0); \
   * }
   */
  #define  TMC_ADV() {  }

#endif // TMC2130 || TMC2208

// @section L6470
```

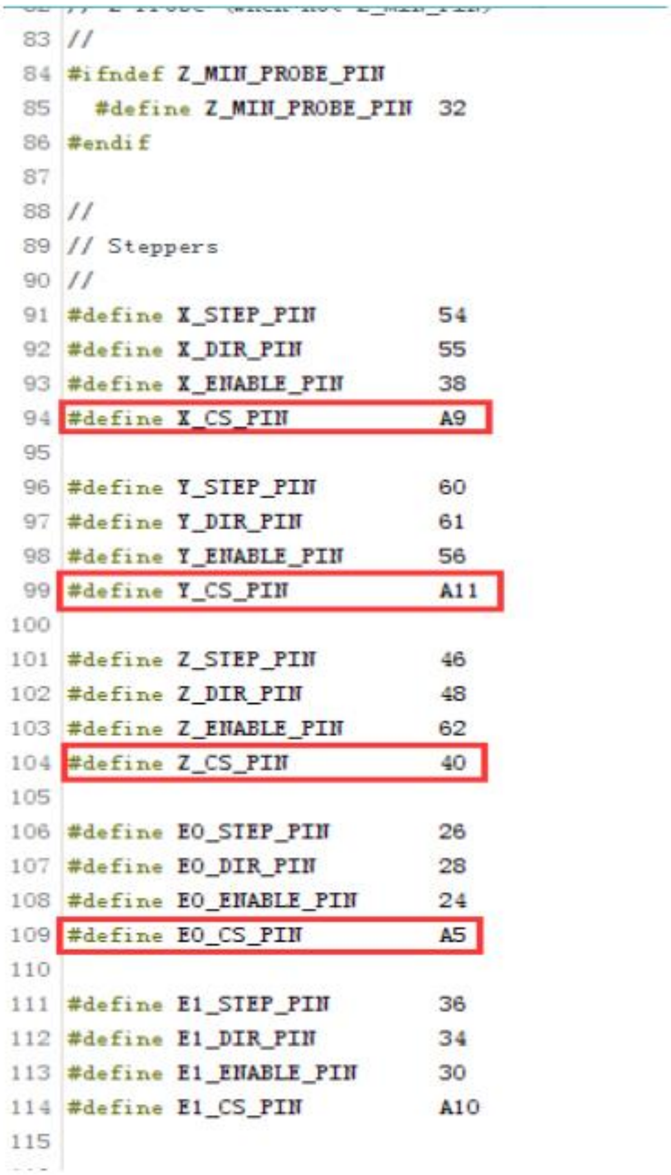
5 is followed by the IO operation to change slice selection CS, open the hide operator in the upper right corner, and drag down until it is found

Pins_ramps.h and search for X_CS



```
// @section TMC2130, TMC2208

/**
 * Enable this for SilentStepStick Trinamic TMC2130 SPI-configurable stepper drivers.
 *
 * You'll also need the TMC2130Stepper Arduino library
 * (https://github.com/teemuatlut/TMC2130Stepper).
 *
 * To use TMC2130 stepper drivers in SPI mode connect your SPI2130 pins to
 * the hardware SPI interface on your board and define the required CS pins
 * in your `pins_MYBOARD.h` file. (e.g., RAMPS 1.4 uses AUX3 pins `X_CS_PIN 53`, `Y_CS_PIN 49`, etc.).
 */
#define HAVE_TMC2130

/**
 * Enable this for SilentStepStick Trinamic TMC2208 UART-configurable stepper drivers.
 * Connect #_SERIAL_TX_PIN to the driver side PDN_UART pin.
 * To use the reading capabilities, also connect #_SERIAL_RX_PIN
 * to #_SERIAL_TX_PIN with a 1K resistor.
 * The drivers can also be used with hardware serial.
 *
 * You'll also need the TMC2208Stepper Arduino library
 * (https://github.com/teemuatlut/TMC2208Stepper).
 */
//#define HAVE_TMC2208

#if ENABLED(HAVE_TMC2130) || ENABLED(HAVE_TMC2208)

  // CHOOSE YOUR MOTORS HERE, THIS IS MANDATORY
  #define X_IS_TMC2130
  //#define X2_IS_TMC2130
  #define Y_IS_TMC2130
  //#define Y2_IS_TMC2130
  //#define Z_IS_TMC2130
  //#define Z2_IS_TMC2130
  //#define E0_IS_TMC2130
```

6. Change IO port according to your requirements:



```
//
#ifndef Z_MIN_PROBE_PIN
  #define Z_MIN_PROBE_PIN  32
#endif

//
// Steppers
//
#define X_STEP_PIN         54
#define X_DIR_PIN          55
#define X_ENABLE_PIN       38
#define X_CS_PIN           A9

#define Y_STEP_PIN         60
#define Y_DIR_PIN          61
#define Y_ENABLE_PIN       56
#define Y_CS_PIN           A11

#define Z_STEP_PIN         46
#define Z_DIR_PIN          48
#define Z_ENABLE_PIN       62
#define Z_CS_PIN           40

#define E0_STEP_PIN        26
#define E0_DIR_PIN         28
#define E0_ENABLE_PIN      24
#define E0_CS_PIN          A5

#define E1_STEP_PIN        36
#define E1_DIR_PIN         34
#define E1_ENABLE_PIN      30
#define E1_CS_PIN          A10
```

IO port selection explanation: find AUX-2 ports on your motherboard and select corresponding ports according to your own needs

IO port. IO port IO port IO port A must start with A character to write in such as A9, A11, etc.

D starts with a number like 40 in the figure above)

AUX-2

GND	A9 D63	D40	D42	A11 D65
5V	A5 D59	A10 D64	D44	A12 D66

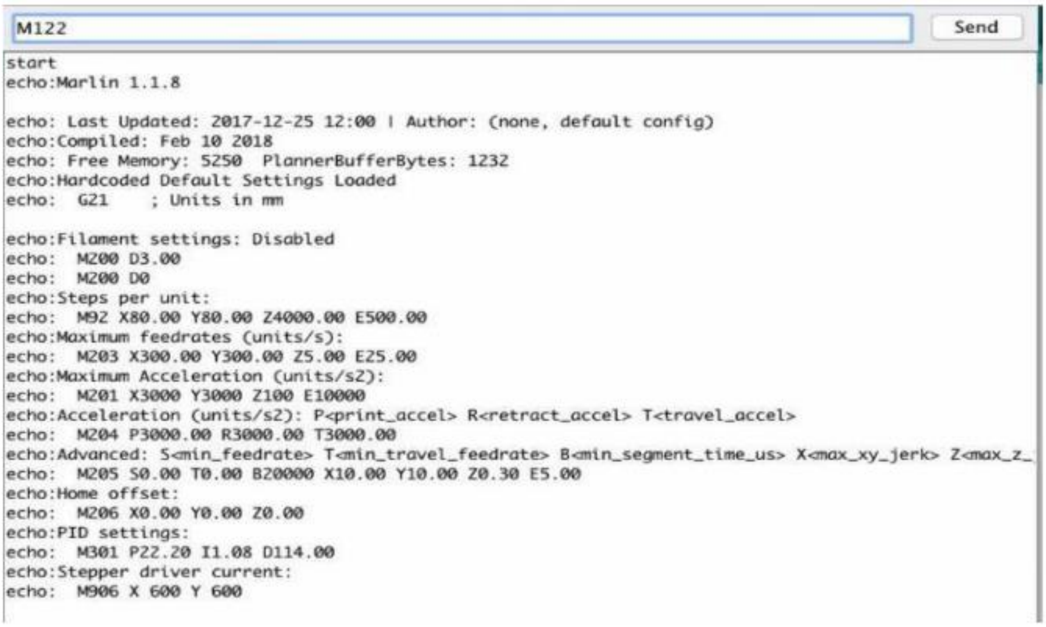
As shown in the IO port diagram above, when writing firmware, you can choose to write A+

number or directly write A number, such as A9

It's the same thing as 63, it's the same IO port. (A11 is equivalent to 65, A12 is equivalent to 66, etc.)

7 after changing IO, do firmware burn and check whether the driver is striving for the link: Ctrl+Shift+M

Open the serial monitor and input M122 to see the detection result:



```
start
echo:Marlin 1.1.8

echo: Last Updated: 2017-12-25 12:00 | Author: (none, default config)
echo:Compiled: Feb 10 2018
echo: Free Memory: 5250  PlannerBufferBytes: 1232
echo:Hardcoded Default Settings Loaded
echo:  G21    ; Units in mm

echo:Filament settings: Disabled
echo:  M200 D3.00
echo:  M200 D0
echo:Steps per unit:
echo:  M92 X80.00 Y80.00 Z4000.00 E500.00
echo:Maximum feedrates (units/s):
echo:  M203 X300.00 Y300.00 Z5.00 E25.00
echo:Maximum Acceleration (units/s2):
echo:  M201 X3000 Y3000 Z100 E10000
echo:Acceleration (units/s2): P<print_accel> R<retract_accel> T<travel_accel>
echo:  M204 P3000.00 R3000.00 T3000.00
echo:Advanced: S<min_feedrate> T<min_travel_feedrate> B<min_segment_time_us> X<max_xy_jerk> Z<max_z_
echo:  M205 S0.00 T0.00 B20000 X10.00 Y10.00 Z0.30 E5.00
echo:Home offset:
echo:  M206 X0.00 Y0.00 Z0.00
echo:PID settings:
echo:  M301 P22.20 I1.08 D114.00
echo:Stepper driver current:
echo:  M906 X 600 Y 600
```