Connecting TMC2130 to trigorilla boards Anycubic kossel

TosikDelta 13.01.2018 15169 21 prints on Anycubic Kossel linear plus

TECHNICAL

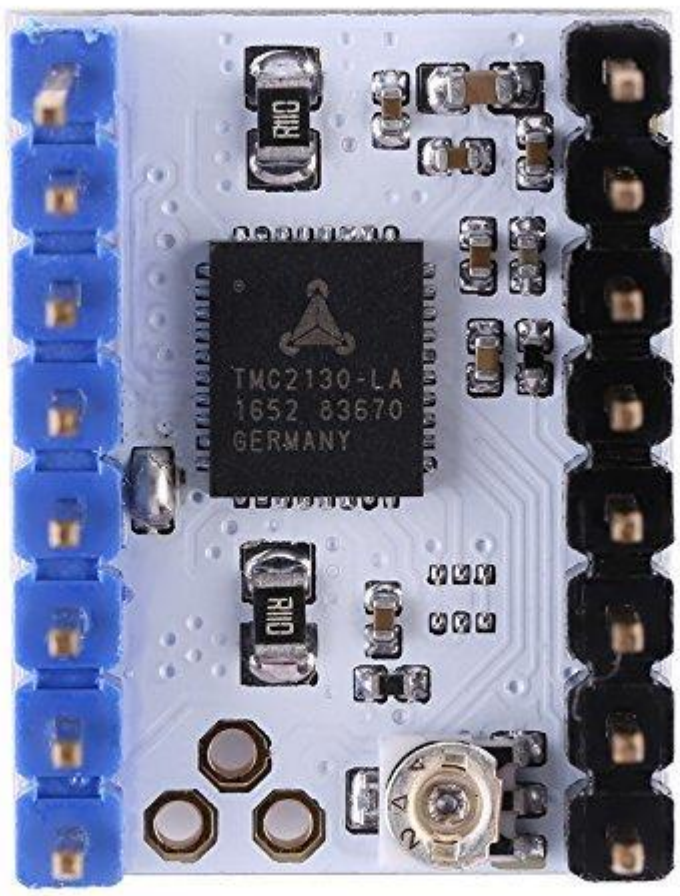
20

This article applies to printers: Anycubic Kossel linear plus

Hello visiting 3d printers and not only.

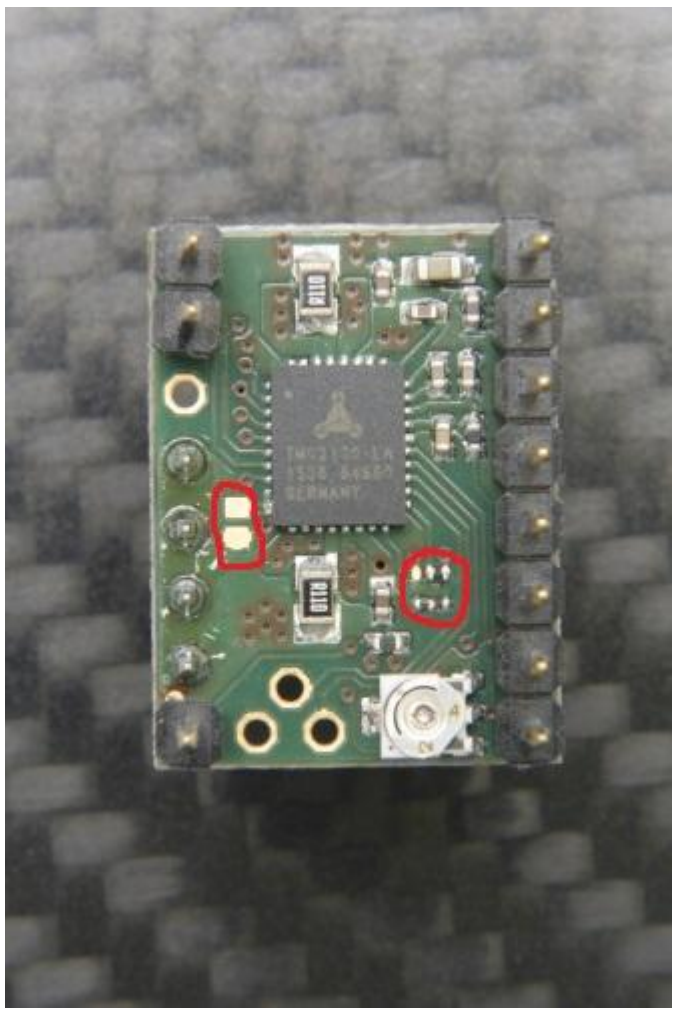
I did not write posts, do not kick much.

Send me Chinese tmc2130,

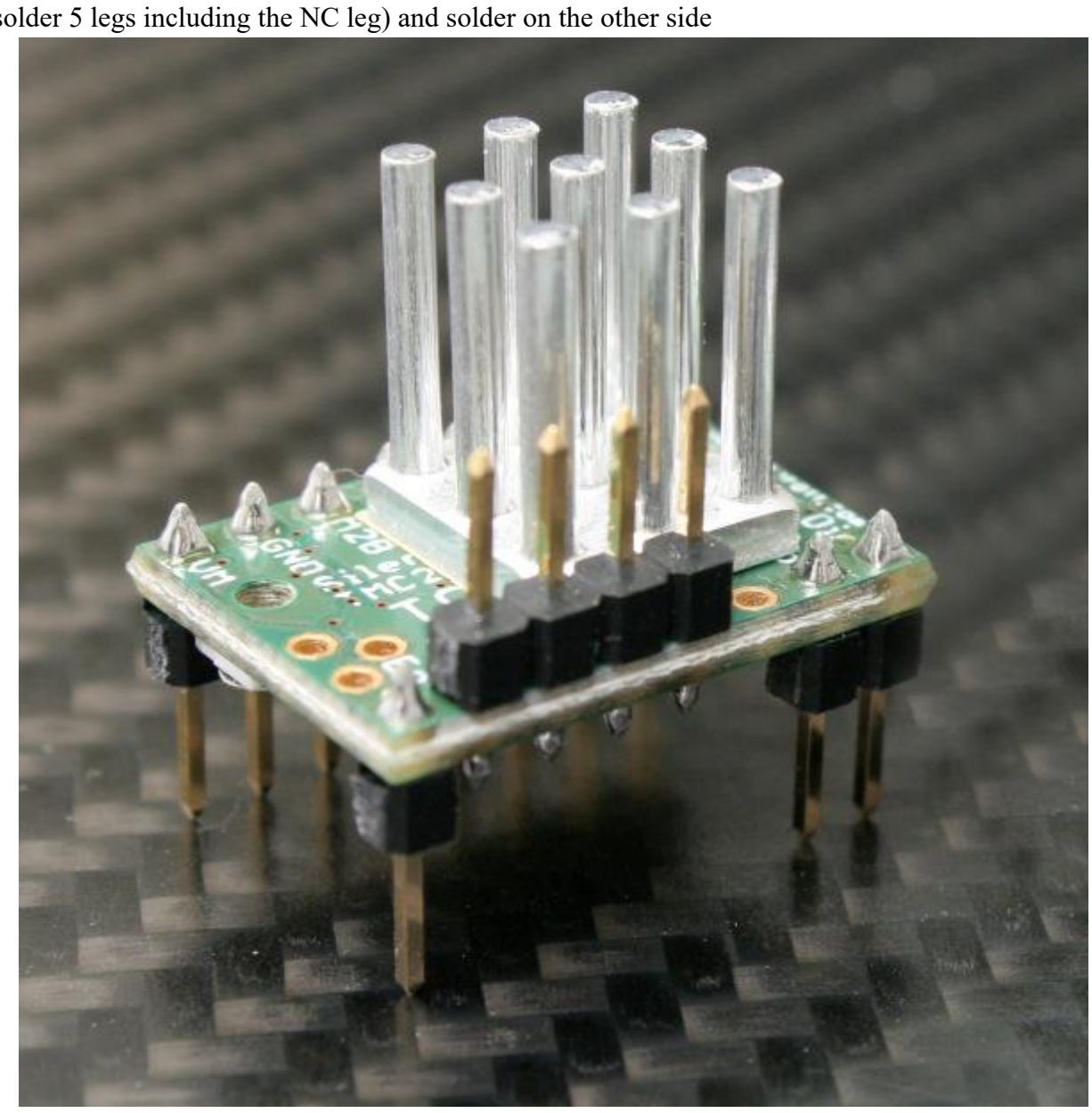


for about a week I was printing with a soldered SPI, I was looking in the open spaces how to connect to Trigorilla, unfortunately there is no such information ... I figured it out myself.

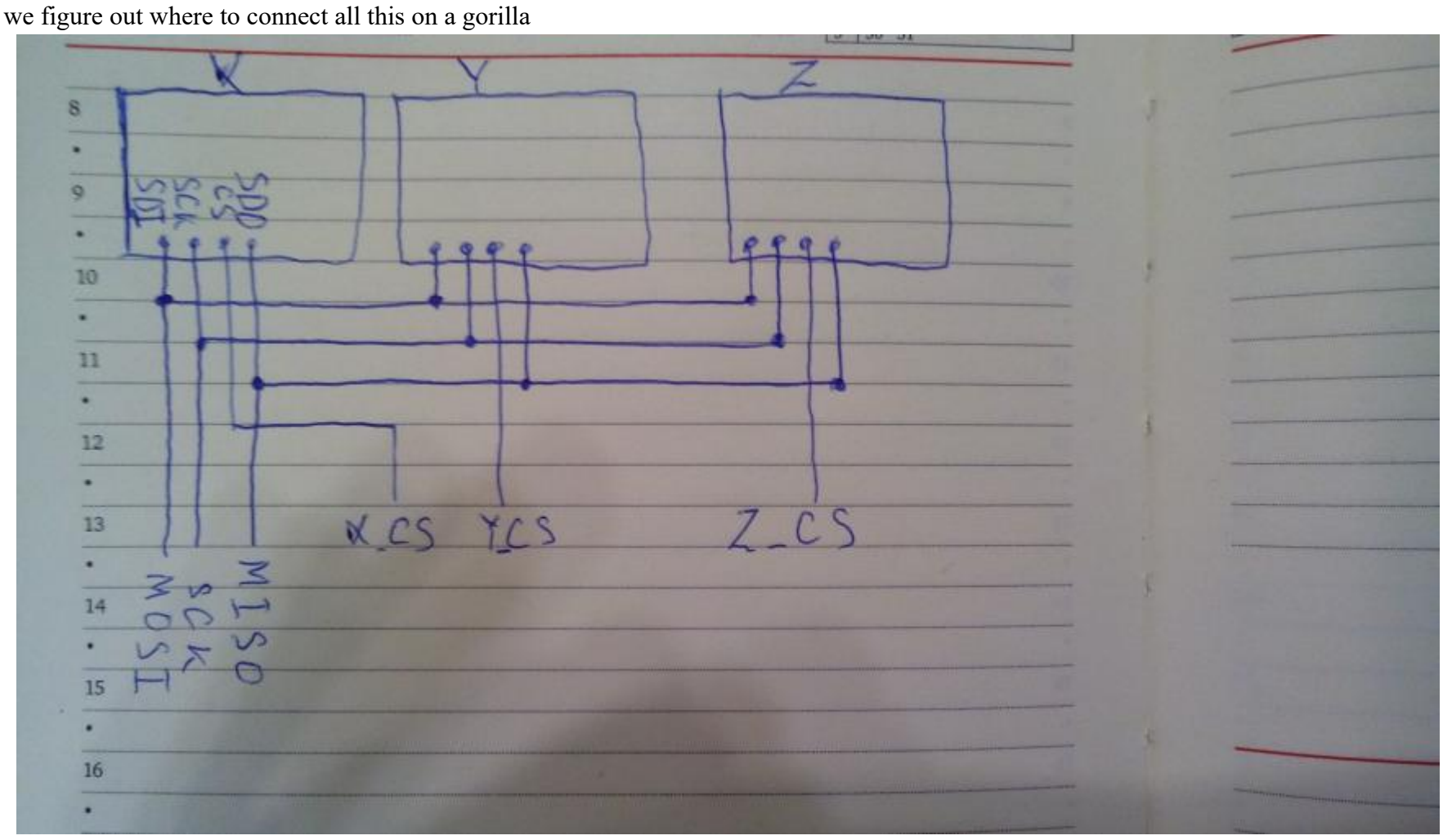
We bring jumpers to this form

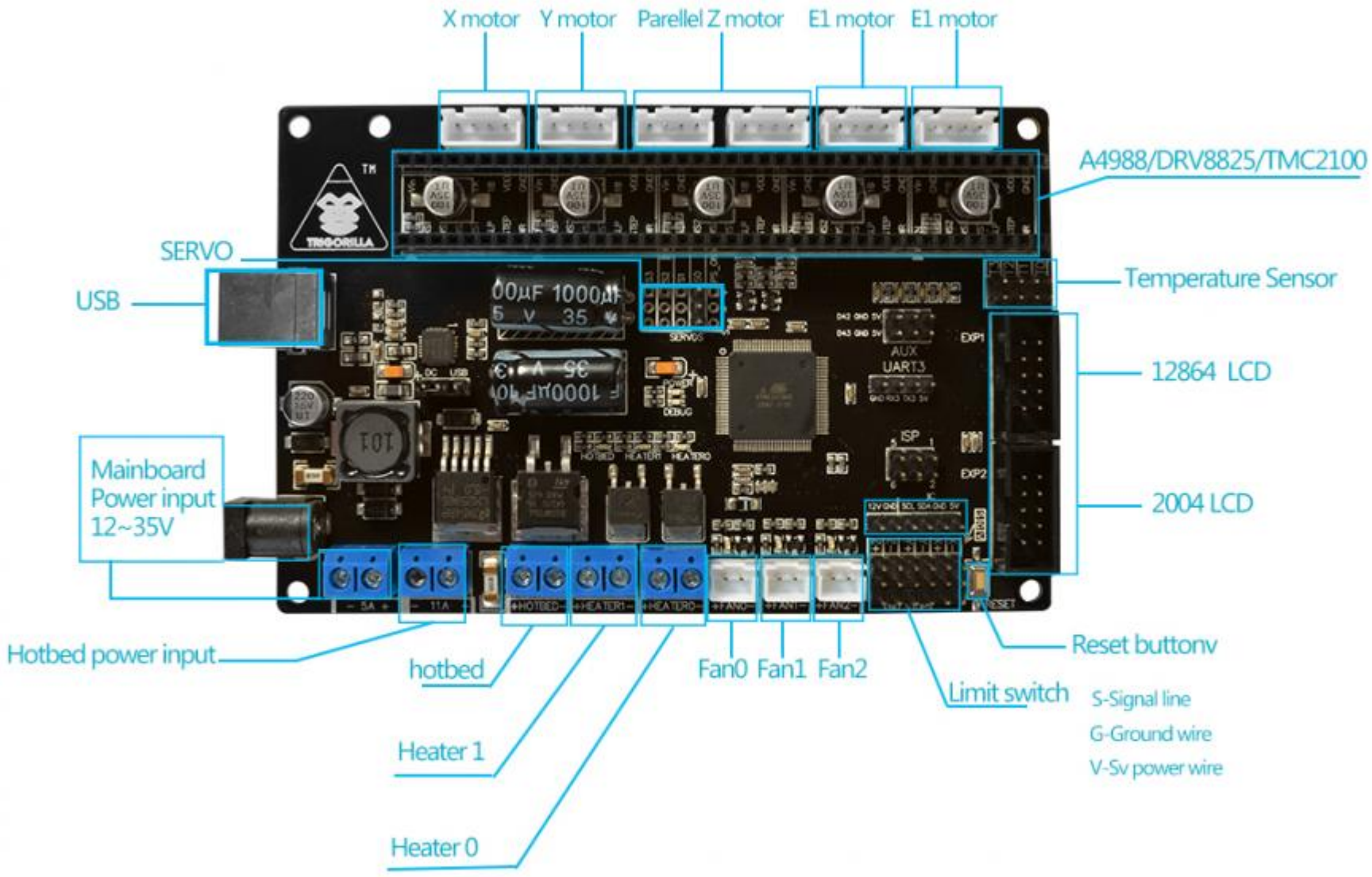


we solder 4 legs (you can solder 5 legs including the NC leg) and solder on the other side



Next, we figure out where to connect all this on a gorilla



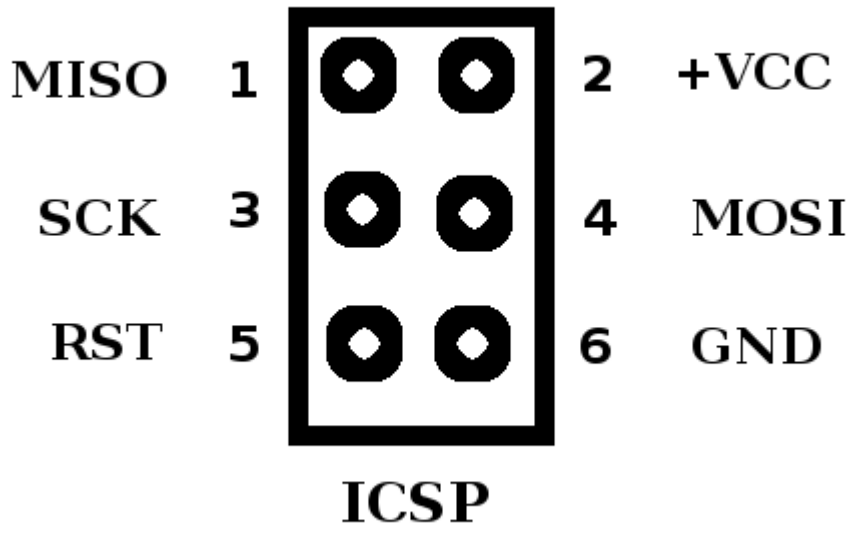


We connect the sdo, sck, sdi contacts to the ISP connector on the

SDO-MISO

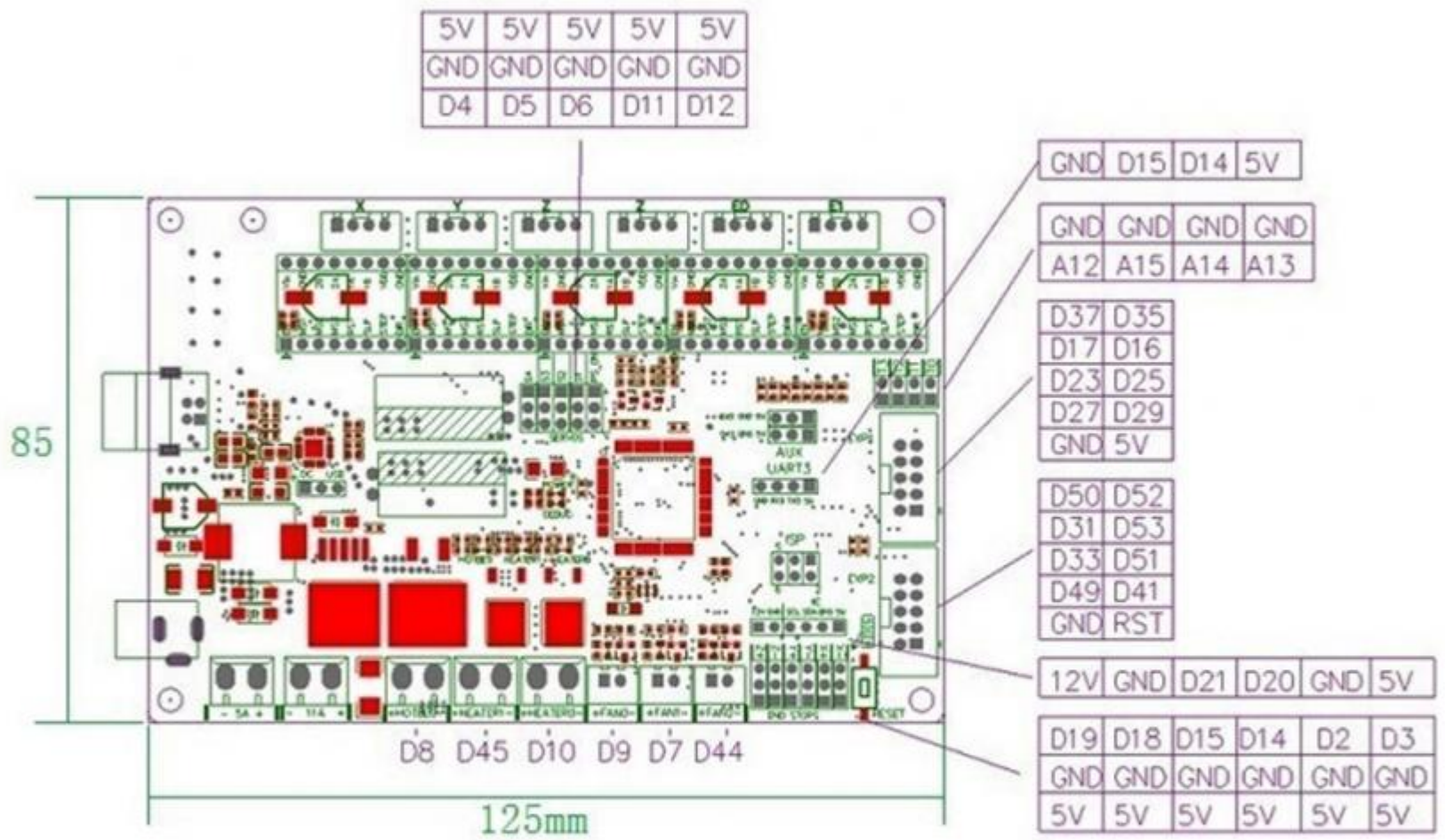
SCK-SCK

SDI-MOSI board



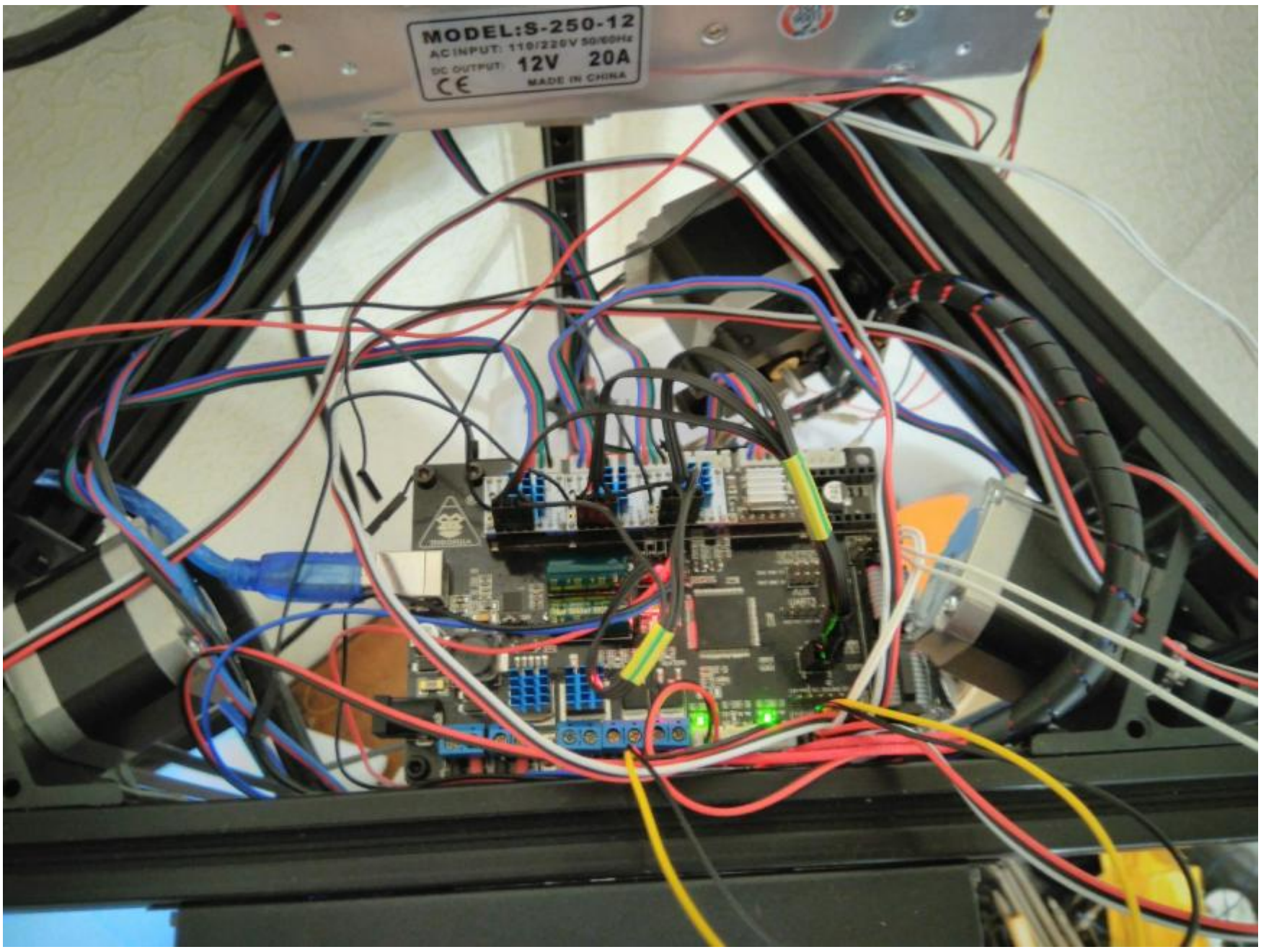
I decided to connect CS contacts from each driver instead of Servo 123 (these are free contacts on my printer)

We look at the picture and see that these are D4D5D6 contacts



we solder the cable, I got this

We install, we connect everything, it turned out something like this



Further, all work in the firmware, you need to enable the use of tmc2130 drivers and select which axis they are on (I have xyz)

Settings in the CONFIGURATION_ADV.H file Enable

driver support

#define HAVE_TMC2130

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

// the CHOOSE YOUR MOTORS HERE, the THIS the IS MANDATORY

#define X_IS_TMC2130

// # the define X2_IS_TMC2130

#define Y_IS_TMC2130

// # the define Y2_IS_TMC2130

#define Z_IS_TMC2130

// # the define Z2_IS_TMC2130

// # the define E0_IS_TMC2130

// # the define E1_IS_TMC2130

// #define E2_IS_TMC2130

// # define E3_IS_TMC2130

// # define E4_IS_TMC2130

Configure current

I have a 1.5 ampere motor, I set the rms current to 900 milliamperes (the maximum current in this case is 1250 milliamperes), the drivers and the motor are almost not heated, there are no skipping steps.

#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 900. // rms current mA in mA Multiply by 1.41 for peak current.

#define X_MICROSTEPS 16 // 0..256

#define Y_CURRENT 900

#define Y_MICROSTEPS 16

#define Z_CURRENT 900

#define Z_MICROSTEPS 16

Turn on STEALTHCHOP

#define STEALTHCHOP

Turn on (As I understand it, the motor overheating control, turned on ...

#define MONITOR_DRIVER_STATUS

Hybrid mode (when a certain speed is exceeded, the driver switches the

microstep mode) #define HYBRID_THRESHOLD

#define X_HYBRID_THRESHOLD_

100Hde

#define Y_HYBRID_THRESHOLD 100

#define Y2_HYBRID_THRESHOLD 100

#define Z_HYBRID_THRESHOLD 3

#define Z2_HYBRID_THRESHOLD 3

#define E0_HYBRID_THRESHOLD 30

#define E1_HYBRID_THRESHOLD 30

#define E2_HYBRID_THRESHOLD 30

#define E3_HYBRID_THRESHOLD 30

#define E4_HYBRID_THRESHOLD 30

For the delta, pay attention to the line

#define Z_HYBRID_THRESHOLD 3

#define Z2_HYBRID_THRESHOLD 3

change the value to 100,

I have this mode turned off. do not print at speeds above 100mms

And vkusnuypirogek for Delta, then, because of what all these drivers bought ...

SENSORLESS_HOMING

Include

#define SENSORLESS_HOMING

Select axis

#if ENABLED (SENSORLESS_HOMING)

#define X_HOMING_SENSITIVITY 8

#define Y_HOMING_SENSITIVITY 8

#endif

As you can see there is no axis Z, this function is not included in the firmware for the z-axis at the moment (Marlin 1.1.8), disappointed, turn it off.

// # define SENSORLESS_HOMING

Turn on

#define TMC_DEBUG

by command M122 outputs information.

Next, go to the pins_RAMPS.h file in the steppers

sections, set the values

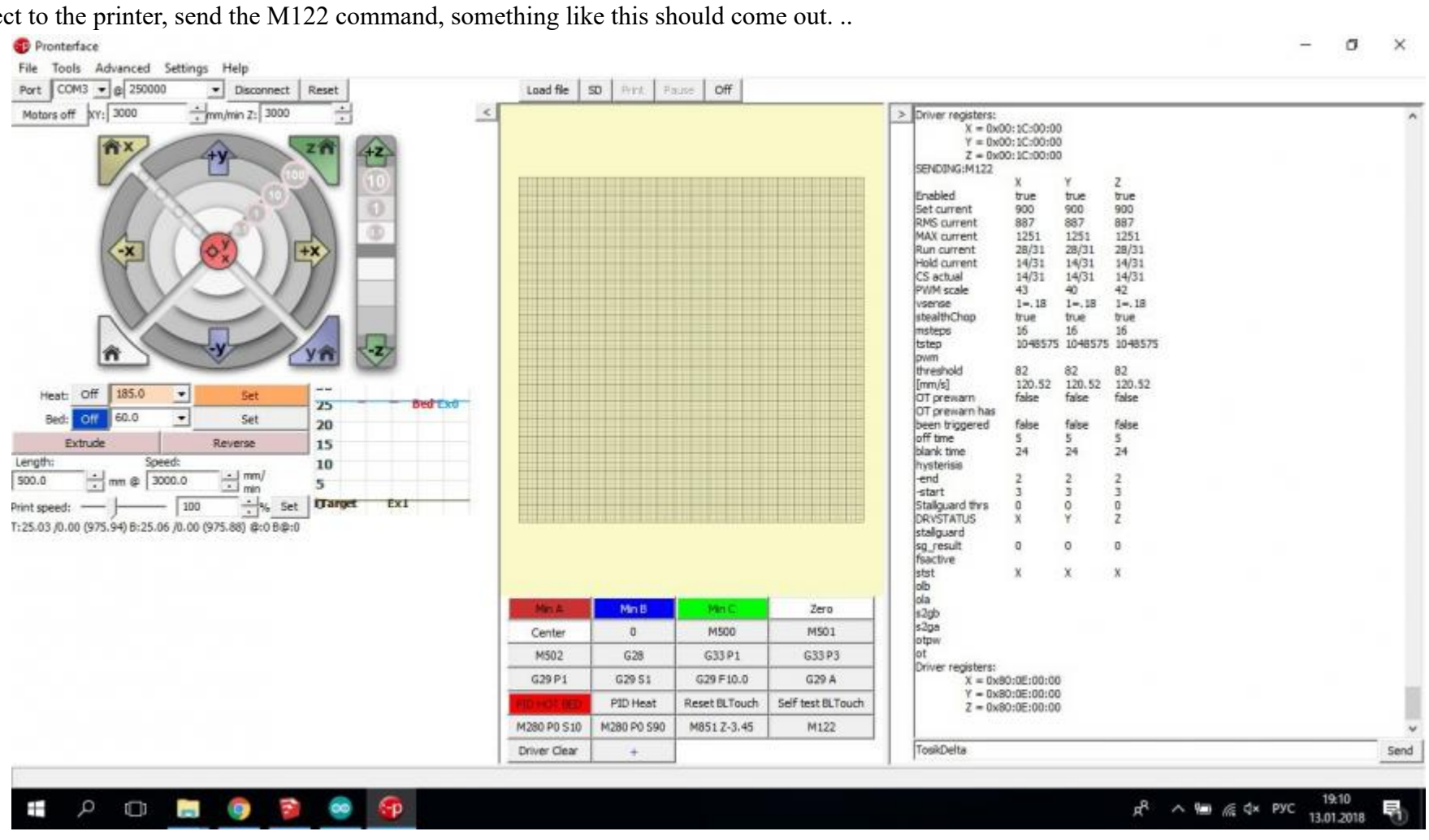
(the CS pin on the driver must be connected to the pin indicated here)

#define X_CS_PIN 4

#define Y_CS_PIN 5

#define Z_CS_PIN 6

Connect to the printer, send the M122 command, something like this should come out. ..



As a result, we got quiet drivers that are controlled by the firmware, but this time it was not possible to get rid of the limit switches.

Perhaps it seemed to me, but it feels like they work quieter with a soldered jumper (without spi control).

Of the losses, 1 burned 2130 and 1 4988

(all due to inattention and not a convenient platform for the driver).

This is a purely informational post and not a call to action, I am not responsible for your burnt equipment.

If possible, I will answer questions in the commentary.