

TMC2130 on trigorilla

Anycubic Kossel

What you need

Common

- Anycubic kossel
- 3 TMC2130 stepstick
- dupont crimps
- pin headers
- cables
- computer
- arduino ide
- working firmware 1.1.7 at least to have debug available

Fysetc 1.0

- solder iron 0.5mm
- magnifying glass
- steady hands

Fysetc 1.1(if I understand right)

- solder iron
- 1 pin

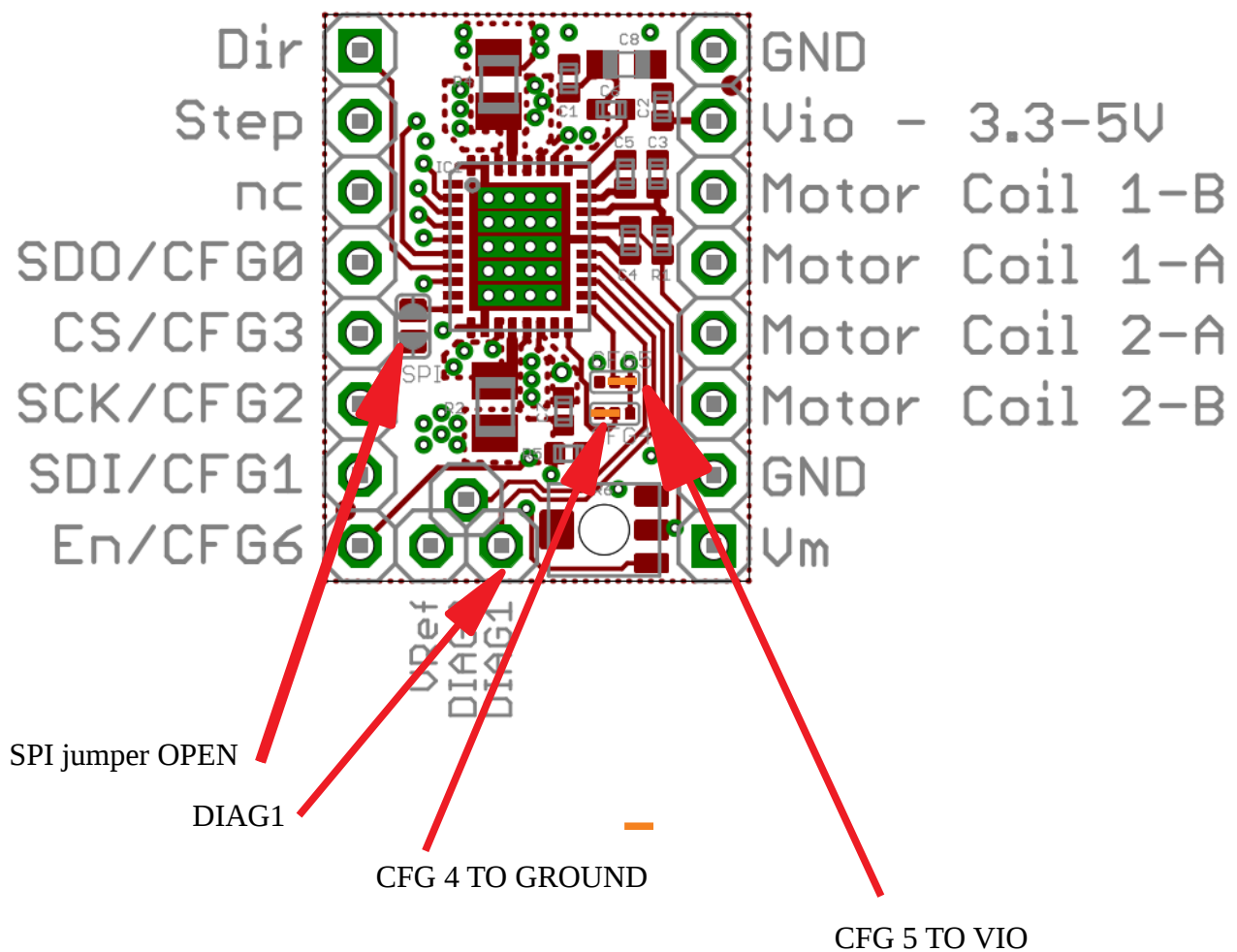
Watterott

- solder iron
- 1 pin

DO AT YOUR OWN RISK

This document is my own notes and I take no responsibility if you do it according to this

First of all check if the stepsticks are configured right



In future firmwares there will be use of the DIAG1 so solder in a pin on that one.
(Sensorless homing)

Make sure that the pins

SDO

CS

SCK

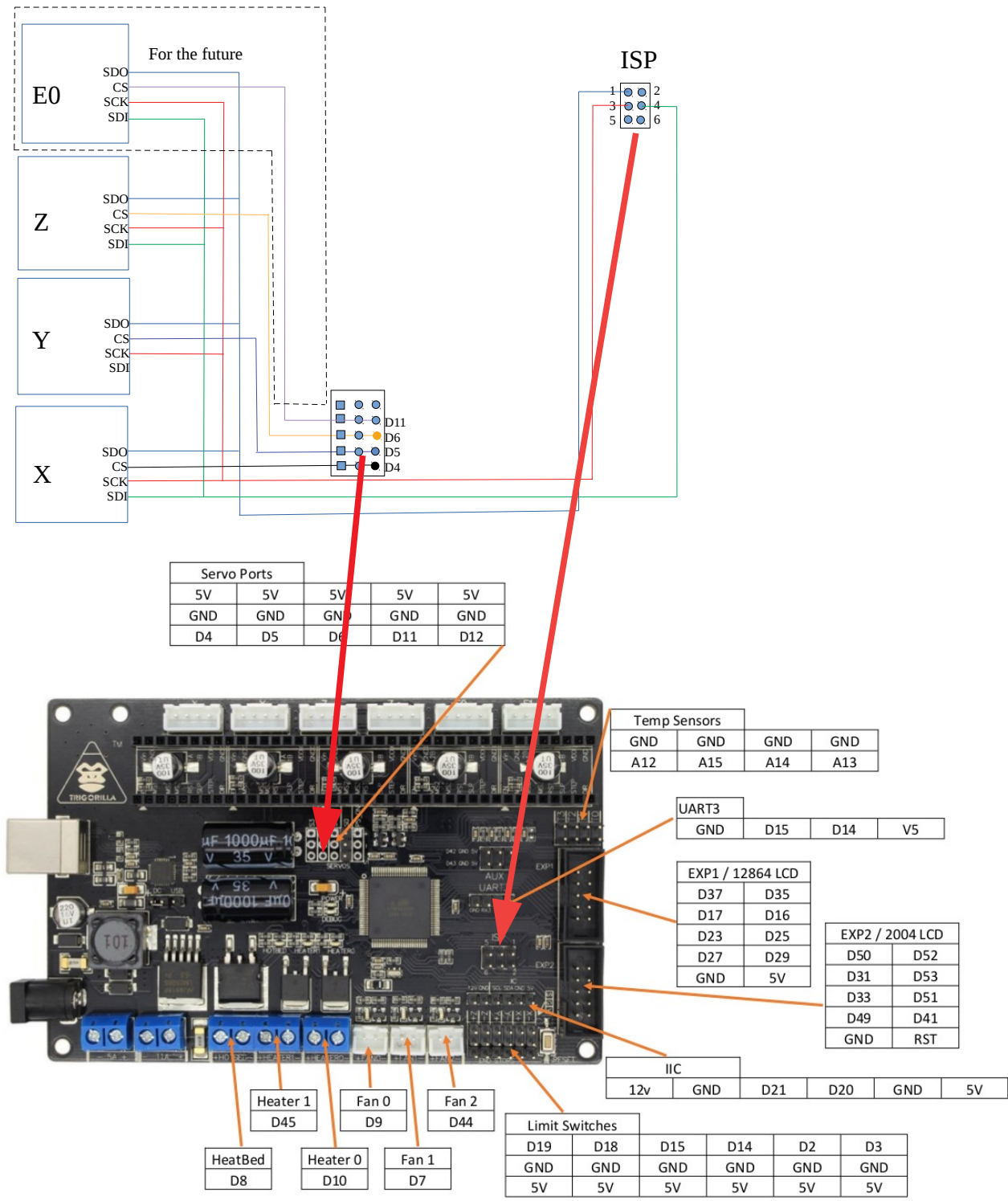
SDI

Is facing upwards

NC isn't connected so it can be left out..

Cables

You need to make a cable harness
like this



Firmware changes

Configuration.h

// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.

#define INVERT_X_DIR **true** //false // DELTA does not invert

#define INVERT_Y_DIR **true** //false

#define INVERT_Z_DIR **true** //false

// For direct drive extruder v9 set to true, for geared extruder set to false.

#define INVERT_E0_DIR false

Configuration_adv.h

// @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 // #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 X_IS_TMC2130
```

```
//define X2_IS_TMC2130
```

```
#define Y_IS_TMC2130 //define Y_IS_TMC2130
```

```
//define Y2_IS_TMC2130
```

```
#define Z_IS_TMC2130 //define Z_IS_TMC2130
```

```
//define Z2_IS_TMC2130
```

```
#define E0_IS_TMC2130 //define E0_IS_TMC2130
```

```
//define E1_IS_TMC2130
```

```
//define E2_IS_TMC2130
```

```
//define E3_IS_TMC2130
```

```
//define E4_IS_TMC2130
```

```
/**
```

- * Stepper driver settings

```
*/
```

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

```
#define X_MICROSTEPS    16 // 0..256
```

```
#define Y_CURRENT      800
```

```
#define Y_MICROSTEPS    16
```

```
#define Z_CURRENT      800
```

```
#define Z_MICROSTEPS    16
```

```
#define E0_CURRENT      1000
```

```
#define E0_MICROSTEPS    16
```

```
/**
```

```
* Use Trinamic's ultra quiet stepping mode.
```

```
* When disabled, Marlin will use spreadCycle stepping mode.
```

```
*/
```

```
#define STEALTHCHOP
```

```
/**
```

```
* Monitor Trinamic TMC2130 and TMC2208 drivers for error conditions,
```

```
* like overtemperature and short to ground. TMC2208 requires hardware serial.
```

```
* In the case of overtemperature Marlin can decrease the driver current until error condition clears.
```

```
* Other detected conditions can be used to stop the current print.
```

* Relevant g-codes:

* M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.

* M911 - Report stepper driver overtemperature pre-warn condition.

* M912 - Clear stepper driver overtemperature pre-warn condition flag.

* M122 S0/1 - Report driver parameters (Requires TMC_DEBUG)

*/

#define MONITOR_DRIVER_STATUS //define MONITOR_DRIVER_STATUS

#if ENABLED(MONITOR_DRIVER_STATUS)

#define CURRENT_STEP_DOWN 50 // [mA]

#define REPORT_CURRENT_CHANGE

#define STOP_ON_ERROR

#endif

/**

* The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.

* This mode allows for faster movements at the expense of higher noise levels.

* STEALTHCHOP needs to be enabled.

* M913 X/Y/Z/E to live tune the setting

*/

#define HYBRID_THRESHOLD //define HYBRID_THRESHOLD

#define X_HYBRID_THRESHOLD 100 // [mm/s]

#define X2_HYBRID_THRESHOLD 100

#define Y_HYBRID_THRESHOLD 100

#define Y2_HYBRID_THRESHOLD 100

#define Z_HYBRID_THRESHOLD **100** //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

/**
 * Use stallGuard2 to sense an obstacle and trigger an endstop.
 * You need to place a wire from the driver's DIAG1 pin to the X/Y endstop pin.
 * X and Y homing will always be done in spreadCycle mode.
 *
 * X/Y_HOMING_SENSITIVITY is used for tuning the trigger sensitivity.
 * Higher values make the system LESS sensitive.
 * Lower value make the system MORE sensitive.
 * Too low values can lead to false positives, while too high values will collide the axis without
triggering.
 * It is advised to set X/Y_HOME_BUMP_MM to 0.
 * M914 X/Y to live tune the setting
 */
//#define SENSORLESS_HOMING // TMC2130 only

#if ENABLED(SENSORLESS_HOMING)
  #define X_HOMING_SENSITIVITY 8
  #define Y_HOMING_SENSITIVITY 8
#endif

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

```



```

*/

#define TMC_DEBUG // #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

```

Ad TMC2130Stepper Arduino library

sketch → include library → Manage library → search after 2130 and install

<https://youtu.be/sPvTB3irCxQ?t=10m43s>

RAMPS.h

```

//

// Steppers

```

```
//  
  
#define X_STEP_PIN    54  
#define X_DIR_PIN     55  
#define X_ENABLE_PIN  38  
#define X_CS_PIN      4 //53  
  
#define Y_STEP_PIN    60  
#define Y_DIR_PIN     61  
#define Y_ENABLE_PIN  56  
#define Y_CS_PIN      5 //49  
  
#define Z_STEP_PIN    46  
#define Z_DIR_PIN     48  
#define Z_ENABLE_PIN  62  
#define Z_CS_PIN      6 //40  
  
#define E0_STEP_PIN   26  
#define E0_DIR_PIN    28  
#define E0_ENABLE_PIN 11 //24  
#define E0_CS_PIN     42  
  
#define E1_STEP_PIN   36  
#define E1_DIR_PIN    34  
#define E1_ENABLE_PIN 30  
#define E1_CS_PIN     44
```

Upload firmware and do a M502 → M500 → M501 to reset save and load the new settings....

Gcodes to tinker with

M122 - TMC Debugging

M906 - TMC Motor Current

M911 - TMC OT Pre-Warn Condition

M912 - Clear TMC OT Pre-Warn

M913 - Set Hybrid Threshold Speed

M914 - TMC Bump Sensitivity(we dont use this yet)

Referenses

<http://3dtoday.ru/blogs/tosikdelta/connection-tmc2130-in-trigorilla-boards-anycubic-kossel/>

<https://youtu.be/sPvTB3irCxQ>

<https://github.com/MarlinFirmware/Marlin/issues/8480>

https://www.aliexpress.com/store/product/1pc-MKS-TMC2130-V1-1-Stepstick-Stepper-Motor-Driver-SPI-with-Heat-Sink-Ultra-silent-Excellent/3480083_32849988952.html?spm=2114.12010615.0.0.4f0e14f6LqDUio%20This%20is%20our%20new%20TMC2130%20V1.1%20for%20SPI

<https://github.com/watterott/SilentStepStick>

DO AT YOUR OWN RISK

This document is my own notes and I take no responsibility if you do it according to this