MKS Gen_L controller board

Ⅰ Overview

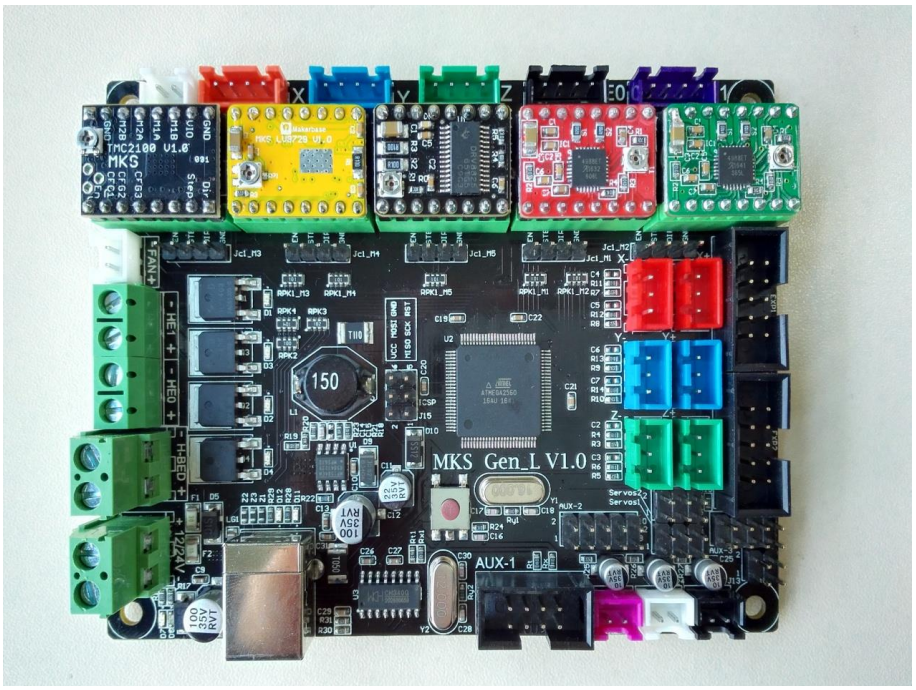
Aiming at the problems exist on the open sourced controller board such as Ramps 1.4, the developers in Makerbase specially optimize and develop a new product, MKS Gen_L, which provides more flexible choices for users.

Ⅱ Features

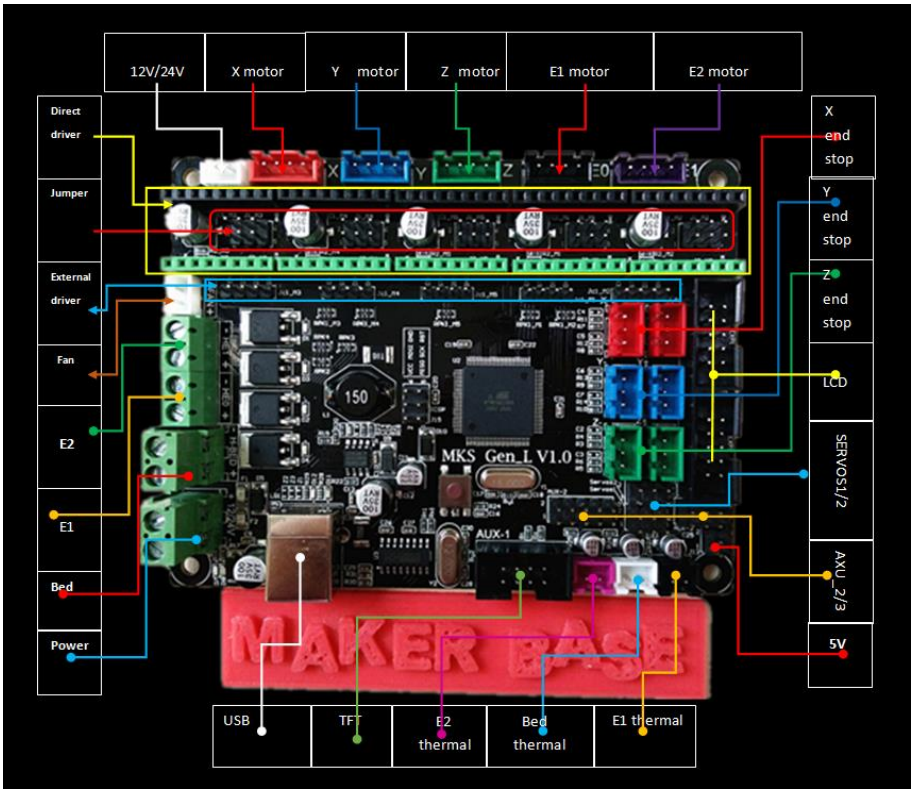
1. 2560 and Ramps 1.4 are integrated in the board, which avoids the complicated connectors.
2. Support a variety of stepper drivers such as A4988, DRV8825, TMC2100 and LV8729, available to change the stepper drivers and motors as users like.
3. Reserve external driver signal, available to connect external high-current driver to drive the 57 and 86 stepper motors.
4. Use high-quality MOSFET, better cooling and more stable.
5. Use specific power chip, support 12V-24V power input, and resolve the problems of chip heating and power shortage.
6. Stable and reliable filter circuit greatly reduces interference and avoids crashes as possible in the process of printing.
7. Use CH340 serial port chip, more easily install the drive than 16U2.
8. Run Marlin firmware, the configuration is completely same as Ramps 1.4, and available to directly replace the Ramps 1.4.
9. Support Ramps 1.4, 2004LCD,12864LCD, TFT28 and TFT32.
10. Available to input 24V, the heated bed current can be reduced to 1/4 at the same system power which effectively resolves the problem that the MOSFET of heated bed heats.
11. X axis, Y axis and Z axis with different color terminal controls the stepper motors and endstops, which is convenient to wire.
12. Plug the stepper drivers according to the different color terminals which decreases the probability of reverse and prevents the controller board damage.

Ⅲ Connection and Dimensions

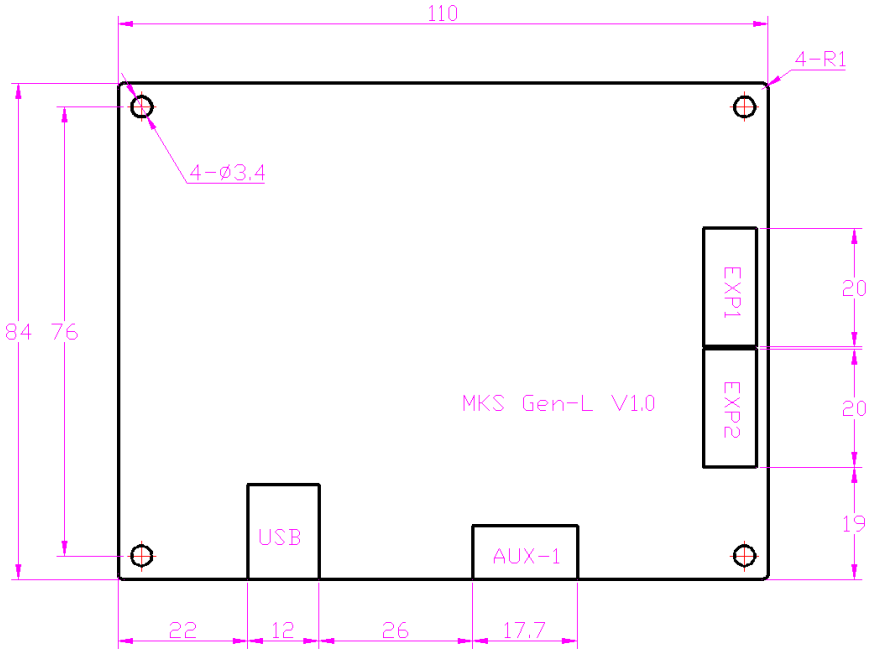
1. Physical Map



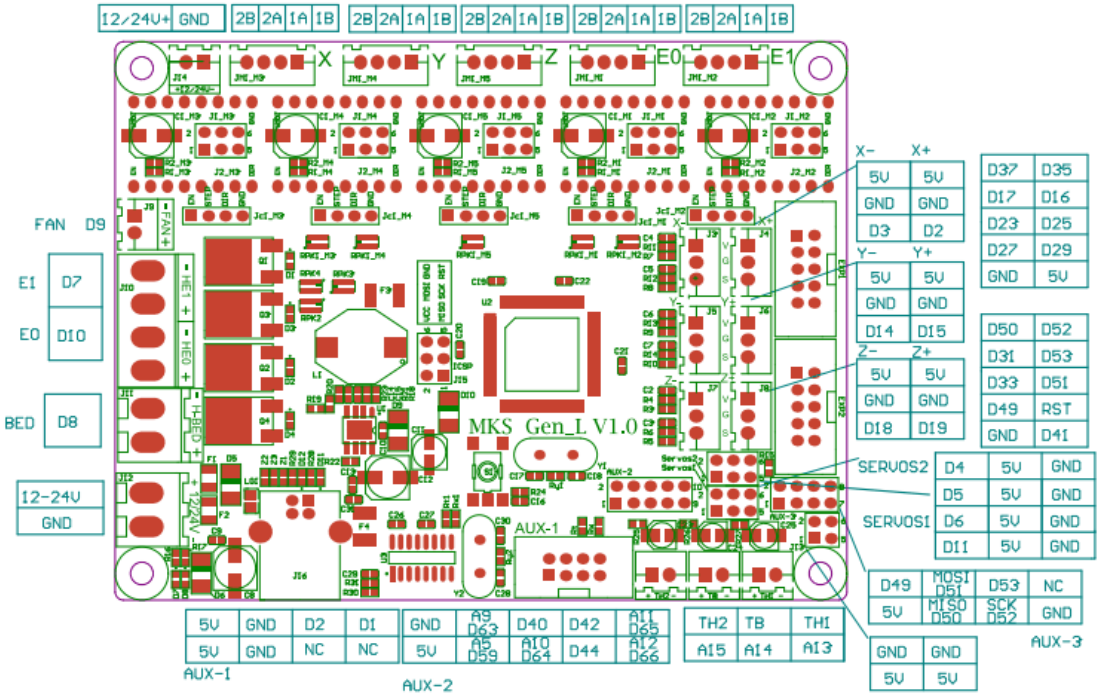
2. System connection diagram



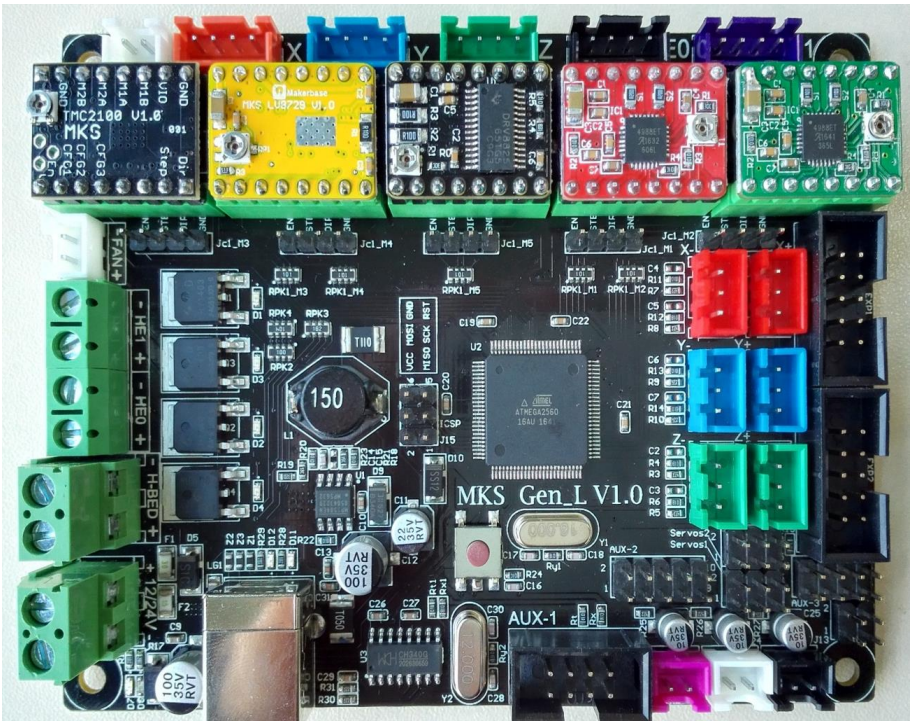
3. Installation Dimensional Drawing



4. PIN port map



5. The connections of different drivers



6. Driver microstepping table

It is M1, M2, M3 in turn from left to right;

Plug the jumper means “High”, pluck the jumper means “Low”.

4988 驱动细分表				8825 驱动细分表				8729 驱动细分表			
M1	M2	M3	细分	M1	M2	M3	细分	M1	M2	M3	细分
Low	Low	Low	Full Step	Low	Low	Low	Full Step	Low	Low	Low	Full Step
High	Low	Low	1/2 Step	High	Low	Low	1/2 Step	High	Low	Low	1/2 Step
Low	High	Low	1/4 Step	Low	High	Low	1/4 Step	Low	High	Low	1/4 Step
High	High	Low	1/8 Step	High	High	Low	1/8 Step	High	High	Low	1/8 Step
High	High	High	1/16 Step	Low	Low	High	1/16 Step	Low	Low	High	1/16 Step
				High	Low	High	1/32 Step	High	Low	High	1/32 Step
				Low	High	High	1/32 Step	Low	High	High	1/64 Step
				High	High	High	1/32 Step	High	High	High	1/128 Step

IV Instructions

1. Acquire the firmware and drive

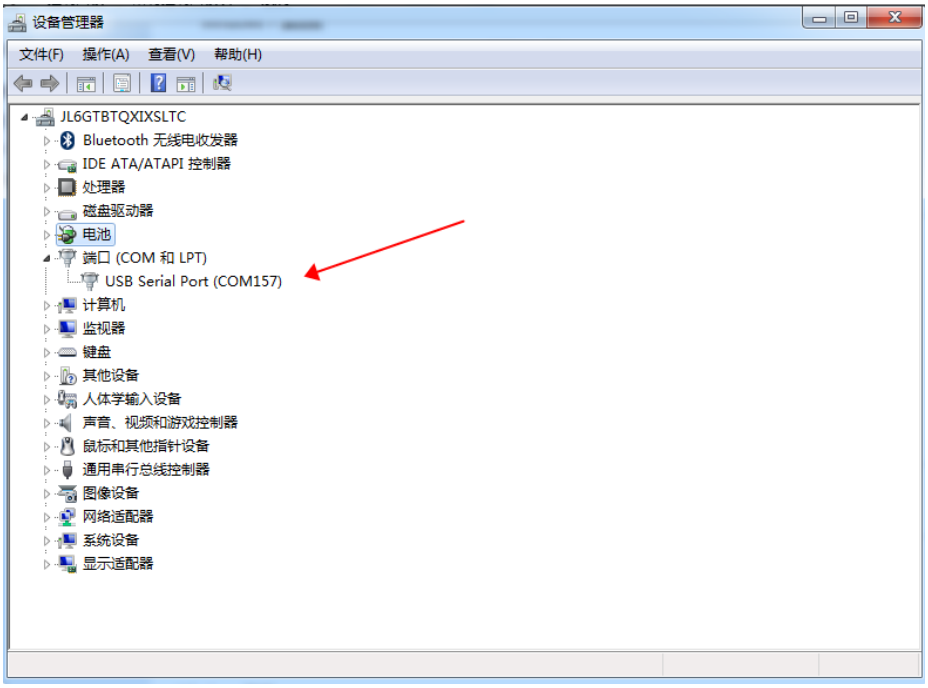
Land the website to download: https://github.com/makerbase-mks?tab=repositories

2. Install the drive

2.1 Find the drive file in the computer, double-click DRVSETUP64 

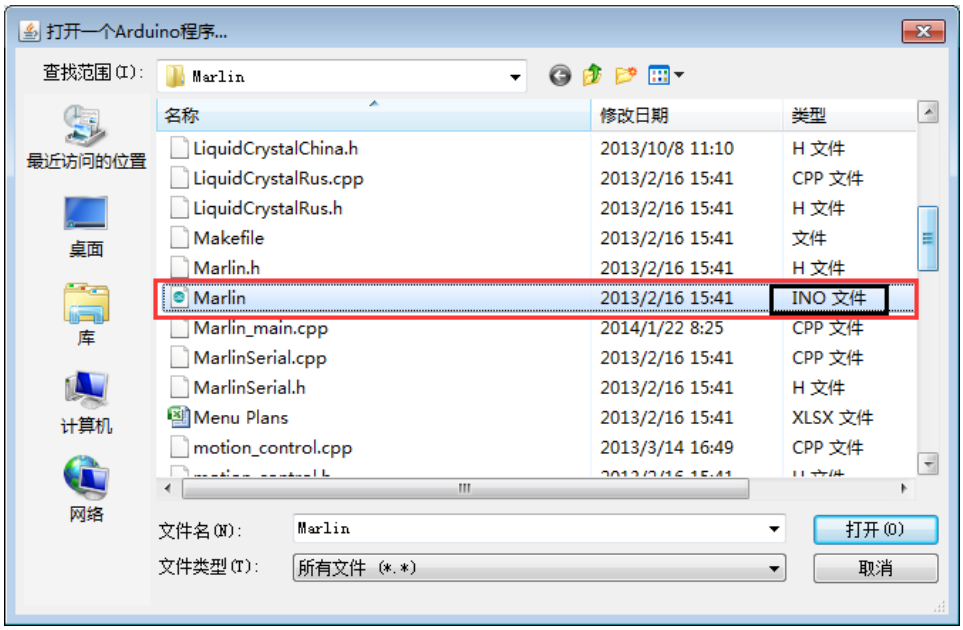
to install the drive.

2.2 After connecting USB, you can see the COM port in the Device Management.

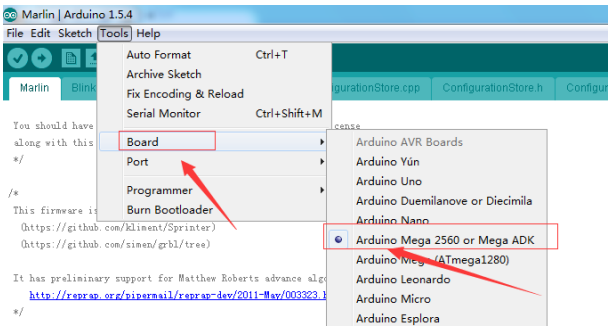


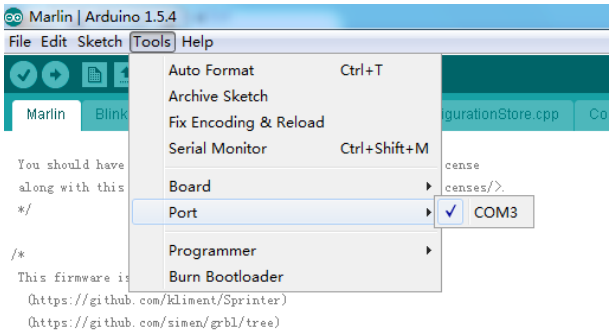
3. Upload the marlin firmware

Firstly, Open Arduino, choose marlin firmware, and select the suffix "***.pde" or "***.ino "file to open.

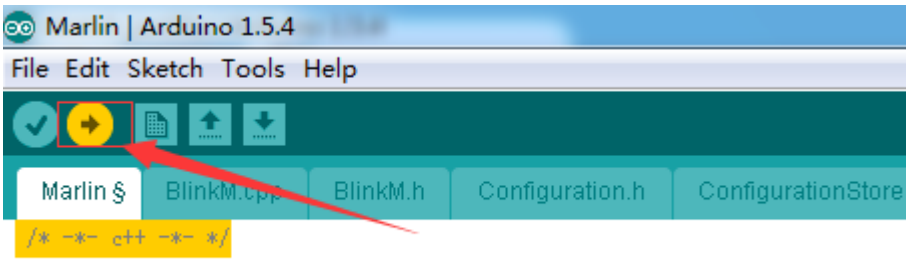


Secondly,



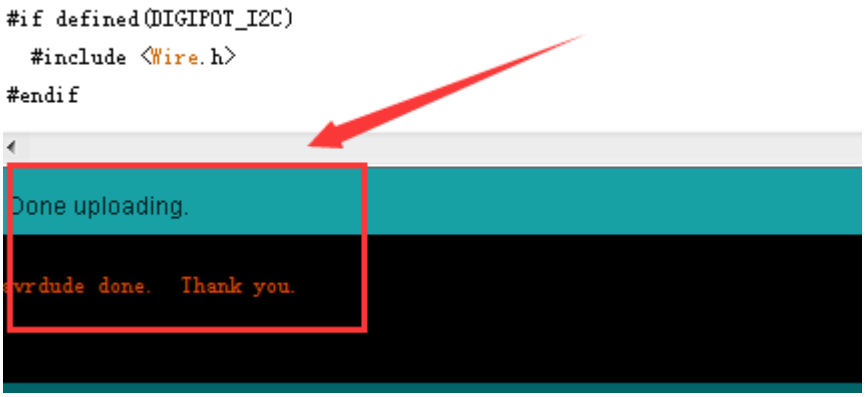


Then,



/*

Finally, finish downloading when the LED on the controller board stop blinking.



Ⅴ Modify the firmware

1. Select the baud rate, the baud rate must be same as the upper monitor's, generally select 115200 or 250000;

```
// This determines the communication speed of the printer
// :[2400, 9600, 19200, 38400, 57600, 115200, 250000]
#define BAUDRATE 250000
```

2. Select the controller board type, select 33 or BOARD_MKS_BASE.

```
// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_MKS_BASE
#endif
```

3. Select the sensor type according to your thermal type. NTC100K thermistor, PT100 thermocouple and AD579 thermocouple is common. If you make use of thermocouple type, you should modify it in the "Pins.h". Take the A11 pin port on the MKS BASE controller board as an example,

```
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_BED 1
```

```
#define TEMP_0_PIN      11   // ANALOG NUMBERING
#define TEMP_1_PIN      15   // ANALOG NUMBERING
#define TEMP_2_PIN      -1   // ANALOG NUMBERING
```

4. There is two types endstop, always-on and always-off.

Always-on mechanical endstop, select "true".

Always-off optical endstop, select "false".

```
// Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
const bool X_MIN_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
const bool Y_MIN_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
const bool Z_MIN_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
const bool X_MAX_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
const bool Y_MAX_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
const bool Z_MAX_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
const bool Z_MIN_PROBE_ENDSTOP_INVERTING = true; // set to true to invert the logic of the endstop.
//#define DISABLE_MAX_ENDSTOPS
//#define DISABLE_MIN_ENDSTOPS
```

5. Control the stepper motors movement direction.

```
// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR true
#define INVERT_Y_DIR false
#define INVERT_Z_DIR false
```

6. Maximum destination of each axis, that is, the printer's largest printing size.

```
// Travel limits after homing (units are in mm)
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS 150
#define Y_MAX_POS 250
#define Z_MAX_POS 200
```

7.Set the pulse number for each axis. The formula for calculating pulse for per motor:

synchronizing wheel: (360/step distance)*microstep/(diameter*3.14)

leadscrew: (360/step)*microstep/lead

```
                                    X  Y  Z  E
#define DEFAULT_AXIS_STEPS_PER_UNIT   {80,80,400,96}  // default steps per unit for Ultimaker
```

```
                                   X  Y  Z  E
#define DEFAULT_AXIS_STEPS_PER_UNIT   {80,80,400,96}  // default steps per unit for Ultimaker
```