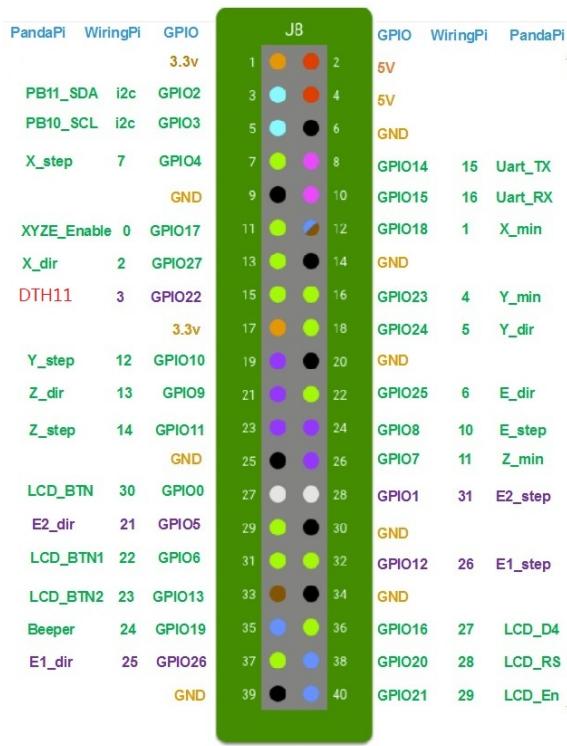


目录

目录	1
Pins-Map	3
pins_PANDA_PI.h	3
Hardware-resources	5
Hardware	5
Support mixed use of 3V-24V fans	5
optocoupler for bed proximity sensor	5
Low temperature mosfet at high current heating.	5
autorecovery fuses and isolation components for over-current and reverse polarity protection for board,drivers,raspberry pi.	5
no wiring for TMC driver sensorless homing and uart mode.	5
Hardware resources v1.0	7
How-to-wire	8
Board connector view	8
1. configure driver :	8
2. connect to LCD128*64:	10
1)connect to Creality UniversalLCD128*64(ender3 CR-10):	11
2)connect to RepRapDiscount Full Graphic Smart Controller	11
3. Fans:	13
How-to-Flash-img-&-WIFI-setup	14
1. RaspberryPi img of PandaPi download link:	14
2. Writing this image to the SD card	14
3. WIFI setup	14
4. Default password	14
5. Shared folder of Pi (only 2021 later image)	15
How-to-Find-Raspberry-IP	16
method 1:Using the phone	16
method 2:Using the laptop or desktop PC	16
How-to-configure-Marlin-code	17
There are two ways to compile marlin:	17
Edit and compile with Browser	17
Compile it with command line(SSH)	17
1. compile source code:	17
2. kill pi_marlin:	18
3. copy the pi_marlin to boot up folder	18
4. run marlin	18
How-to-run-TMC2209-with-UART-mode-(V2.0-2.5)	19
Wiring	19
Attaching the TMC2209 to raspberrypi UART	19
For Pandapi V2.0	19
For Pandapi V2.5	20
Jumpers setting for uart and address of TMC2209, and then plug in TMC2209	20
Configure TMC2209 for Marlin2.0.x	20
click"Save this file" and "Verify", and then "Run"	21
FAQ :	22
Have fun!	22
Run-TMC2208-with-UART-mode	23
Jumpers setting for uart and address of TMC2208	23
Configure TMC2208 for Marlin2.0.x	24
click"Save this file" and "Verify", and then "Run"	24

Have fun!	24
How-to-run-TMC2208-TMC2209-with-standalone-mode	25
Standalone mode for TMC2208/2209	25
How-to-setup-your-own-configuration	26
WARNING !!! - If you do this you won't get updates from https://github.com/markniu/PandaPi anymore and have to merge changes manually	26
How-to-run-OctoPrint	27
Just need to change the 'serial settings' at the first time.	27
How-to-plugin-OctoPrint-Enclosure(DTH11-temperature-humidity-sensor)	29
Wiring:	29
Example:	29
reference link:	30
How-to-connect-to-PT100	31
How-to-install--HDMI-LCD	32
1. Desktop	32
2.Octoscreen	32
How-to-install-desktop(HDMI_LCD)	34
Customize-the-UI-(replace-Image)	35
How-to-wire-BLtouch	36
Wiring	36
Configure software	36
Calibrating BLTouch from LCD display controls (Marlin1.x)	36
How-to-flash-MCU-firmware	38
How-to-use-PID-auto-tune	40
How-to-wire-Proximity-sensor	41
How-to-wiring-filament-runout-sensor	42
PandaPI V2.0	42
PandaPI V2.5	44
Run out sensor Pin:	44
Remap-BED-Hotend0-Hotend1-(-heater_thermistor)	45
upgrade-to-printer-ender5	46
where-to-download--board-case	47
How-to-install-pandapi-to-the-latest-octopi-image	48
1 , compile RT kernel and replace the old one in the image of octopi;	48
2, install tty0tty driver:	48
3, download the marlin compile&configure tools	48
4,download marlin source code	48
5,install pi_marlin	48
6,add the autorun command in the start shell /etc/rc.local	48
7.configure auto mount USB stick	48
FAQ	49
endstop not work?	49
TMC Error message:	49
LCD12864 blank , but can be connected from octopi.	49

Pins-Map



pins_PANDA_PI.h

```

//  

// Steppers  

//  

#define X_STEP_PIN      7  

#define X_DIR_PIN       2  

#define X_ENABLE_PIN    0  
  

#define Y_STEP_PIN      12  

#define Y_DIR_PIN       5  

#define Y_ENABLE_PIN    0  
  

#define Z_STEP_PIN      14  

#define Z_DIR_PIN       13  

#define Z_ENABLE_PIN    0  
  

#define E0_STEP_PIN     10  

#define E0_DIR_PIN      6  

#define E0_ENABLE_PIN   0  
  

#define E1_STEP_PIN     26  

#define E1_DIR_PIN      25  

#define E1_ENABLE_PIN   0  
  

#define E2_STEP_PIN     31  

#define E2_DIR_PIN      21  

#define E2_ENABLE_PIN   0

```

Pins for LCD12864 REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER

```

#define LCD_PINS_RS    28  

#define LCD_PINS_ENABLE 29

```

Pins-Map

```
#define LCD_PINS_D4    27
#define BEEPER_PIN    24
#define BTN_ENC     30
#define BTN_EN1     22
#define BTN_EN2     23
```

For example, if you want to plug the E0 motor line to the E2 connector on the board. just switch the driver pins of E0 and E2 in pins_PANDA_PI.h

before:

```
#define E0_STEP_PIN      10
#define E0_DIR_PIN       6
#define E0_ENABLE_PIN     0

#define E2_STEP_PIN      31
#define E2_DIR_PIN       21
#define E2_ENABLE_PIN     0
```

after:

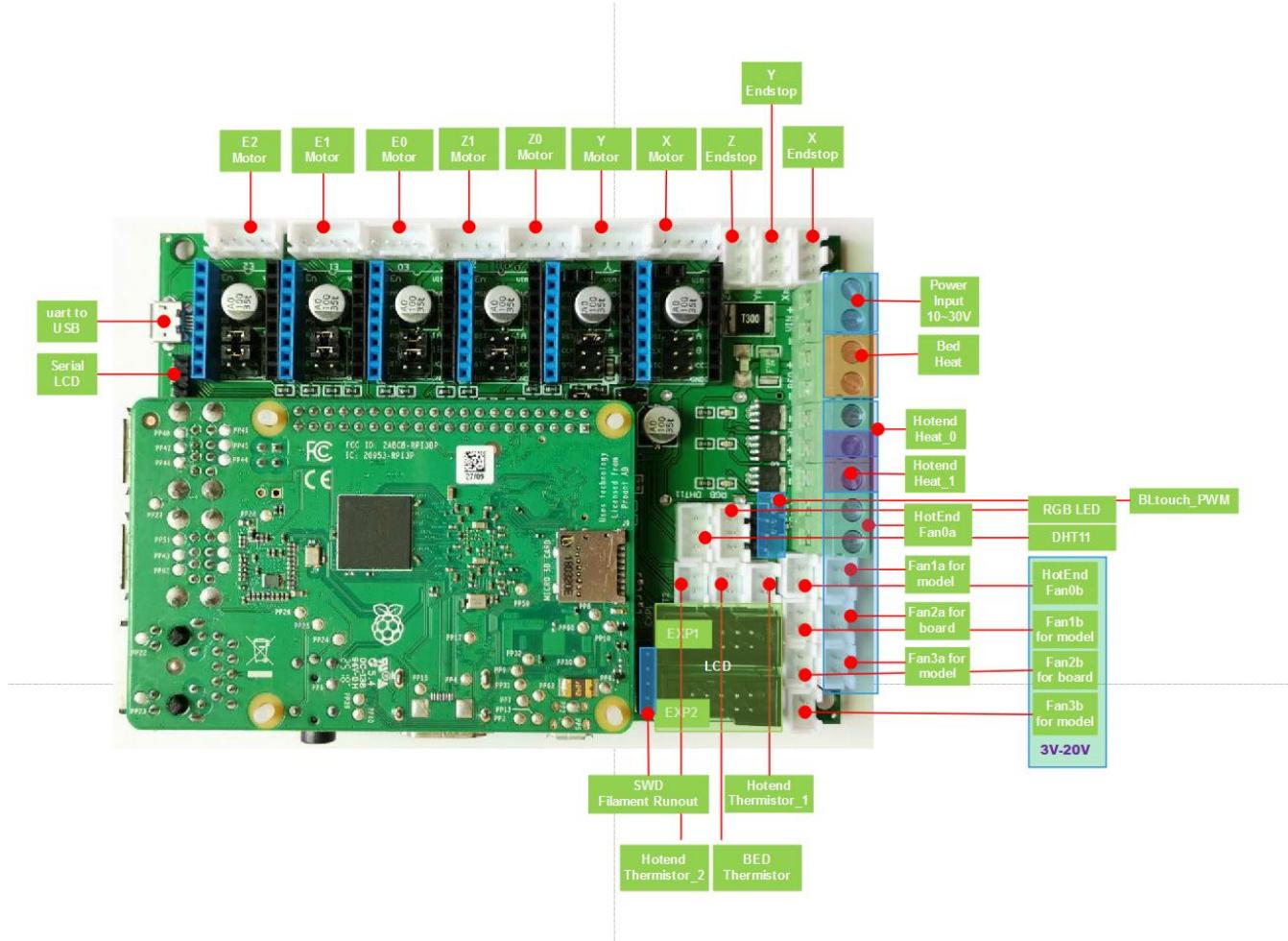
```
#define E0_STEP_PIN      31
#define E0_DIR_PIN       21
#define E0_ENABLE_PIN     0

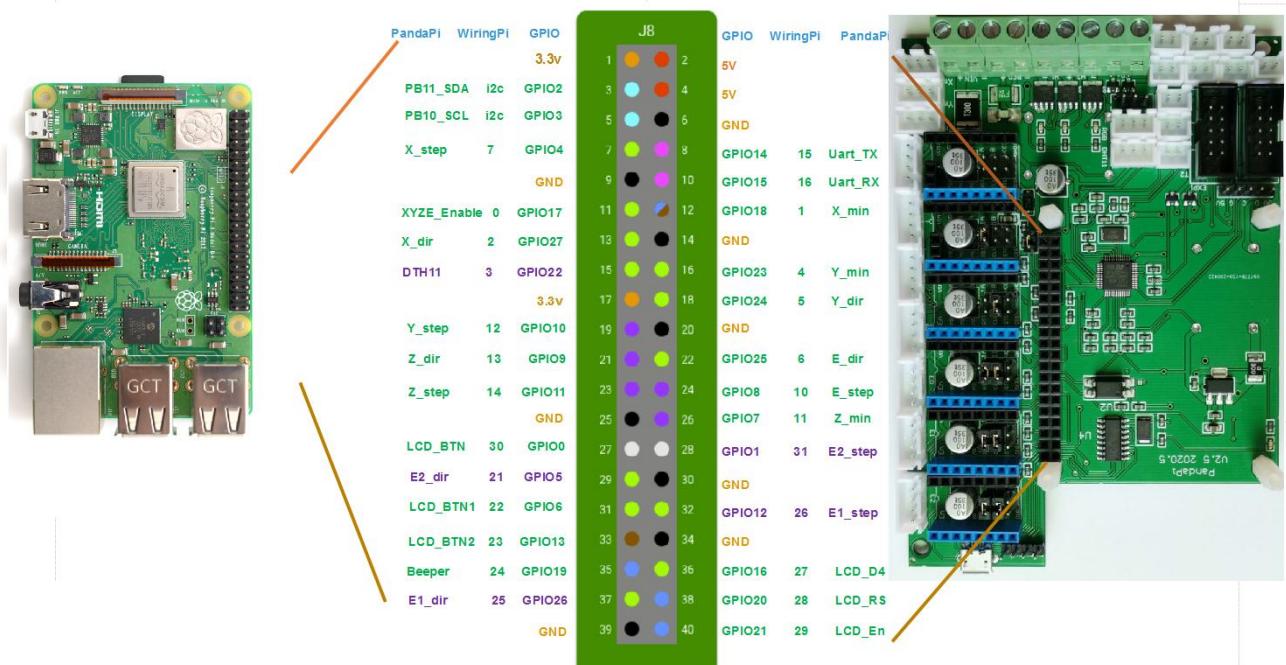
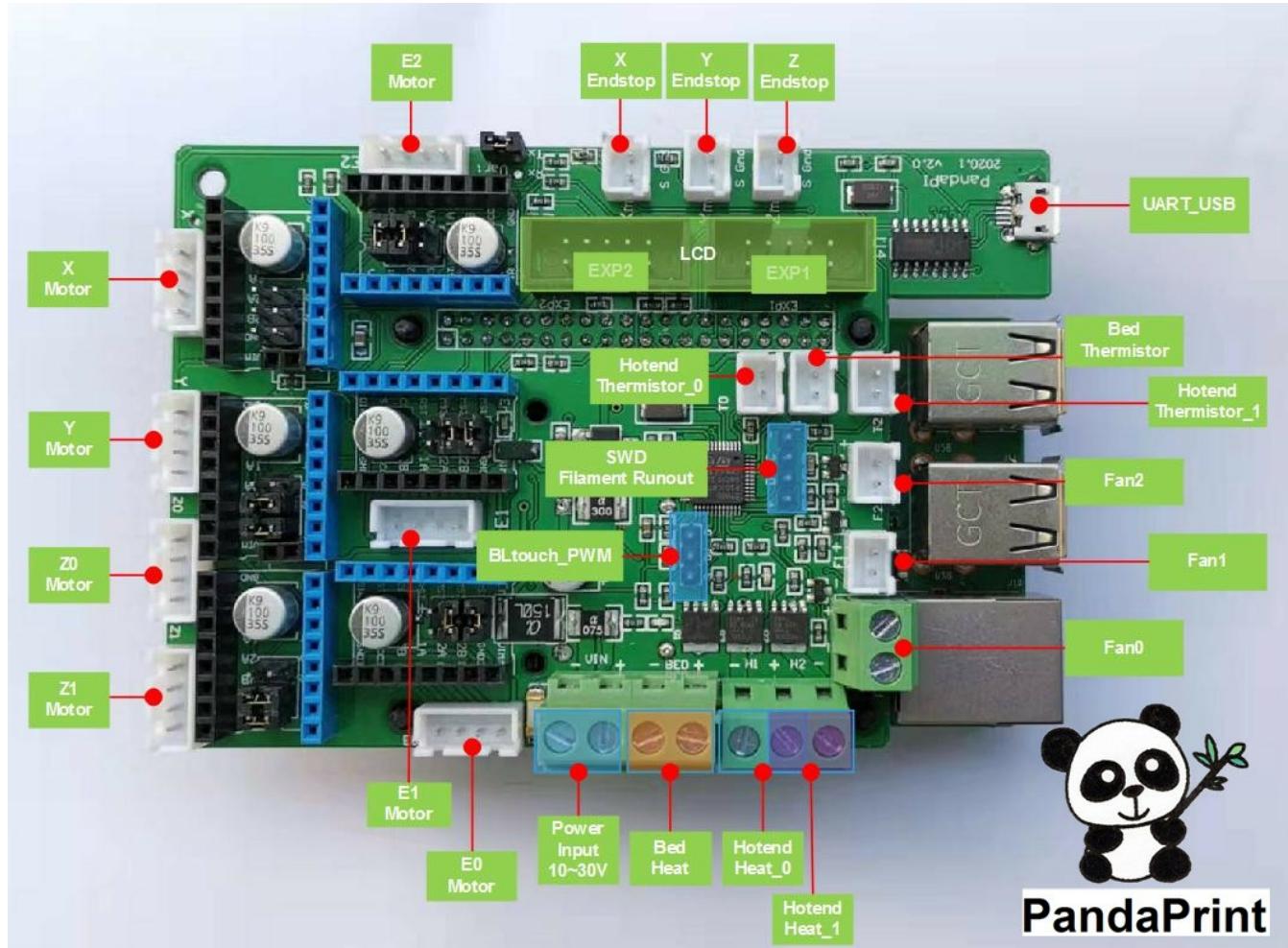
#define E2_STEP_PIN      10
#define E2_DIR_PIN       6
#define E2_ENABLE_PIN     0
```

Hardware-resources

Hardware

1. Support mixed use of 3V-24V fans
2. optocoupler for bed proximity sensor
3. Low temperature mosfet at high current heating.
4. autorecovery fuses and isolation components for over-current and reverse polarity protection for board,drivers,raspberry pi.
5. no wiring for TMC driver sensorless homing and uart mode.





- **Hardware resources v1.0**

RaspberryPi	Pi 4B/3B/3B+	.
Extruders	2	
Controlled Fans	3	FAN0:hotend cooler,it will be opened if the temperature is higher than 60°C.FAN1:model cooler, controlled by the gcode file or gcode command, M106: Set Fan Speed "M106 S255", M107: Fan Off.FAN2:board cooler,If the driver is working, it will be opened.
Heaters	3	
Endstops	3	
Temp sens	3	100K NTC (thermal resistance)
SWD	1	STlinkV2
Serial port chip	CH340G	
stepper driver	6*(TMC2209/TMC2208/A4988)	Modular, replaceable,uart for TMC2209 sensorless endstop
Input	9~28V 20A max	power both the board and the Raspberry Pi
heater Out put	15Amax	
MCU	stm32	
LCD	Graphic12864/CreativityLCD12864/HDMI	
on board FA N	Support mixed use of 3V-24V fans	
Spacer screw	Nylon spacer screw	
filament detect pin	reuse the SWD pin	
Bed leveling	BLtouch	support wild range voltage 5V~36V Inductive sensor NPN, since it have optocoupler.
Protection	4 autorecovery fuses and isolation components	on board for over-current and reverse polarity protection for board,drivers,raspberry pi.

How-to-wire

Board connector view

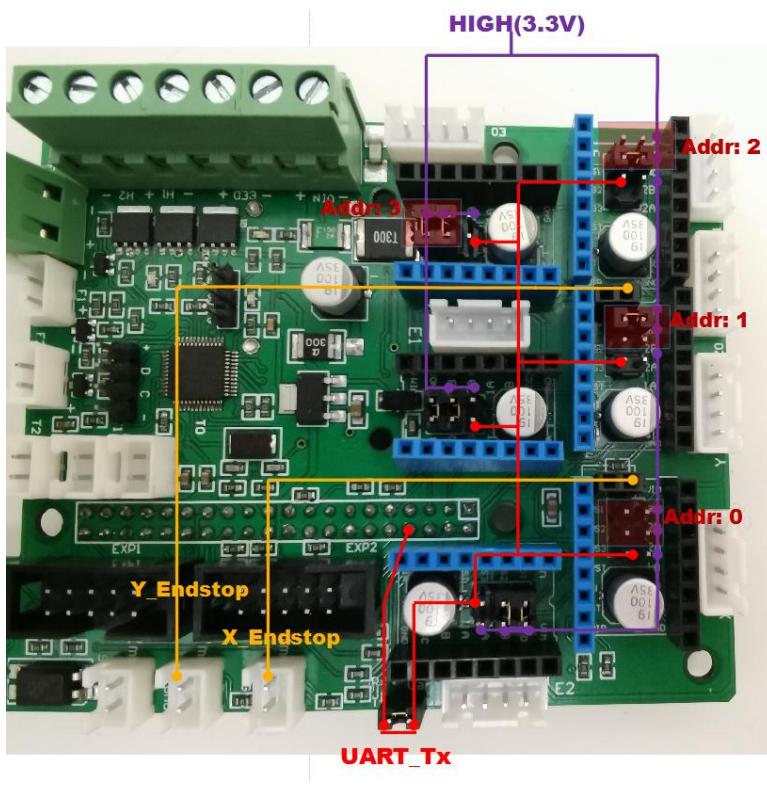
board V2.0

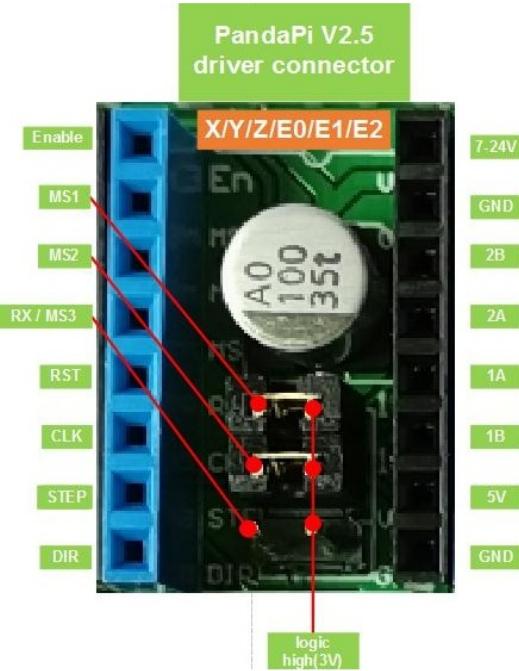
board V2.5

Thanks @Johnny Johnsen he draw this 3D wiring

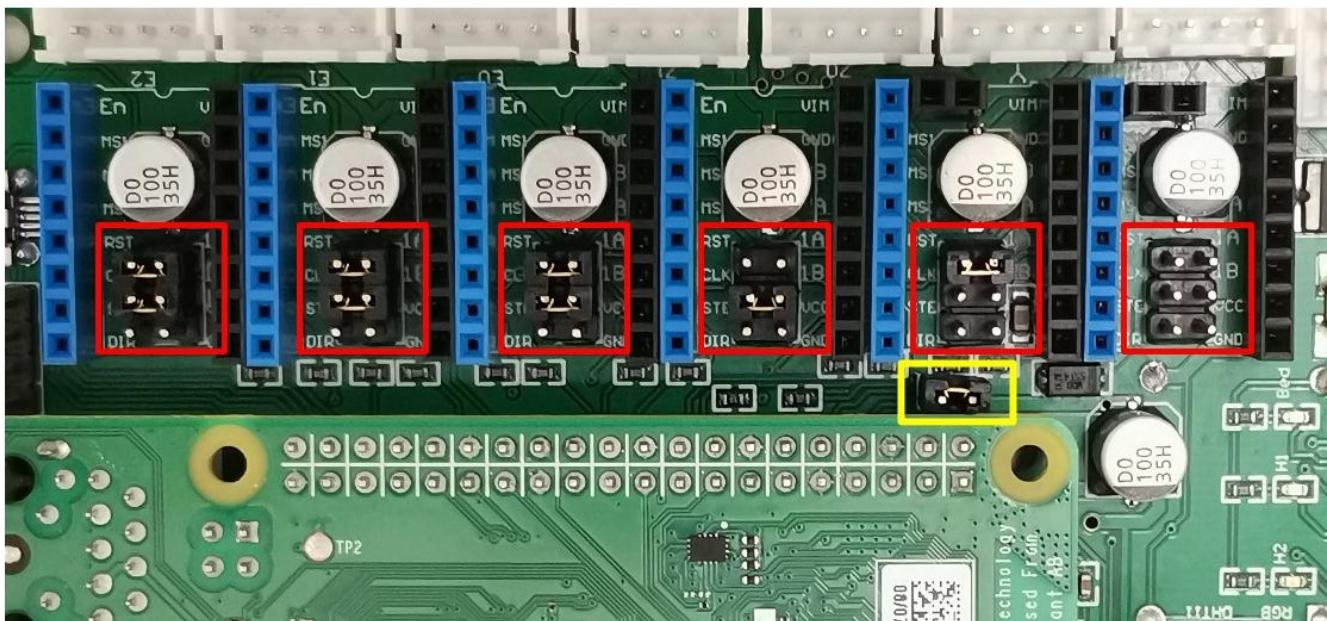


1. configure dirver :





Jumpers for using TMC2209 as UART mode



2. connect to LCD128*64:

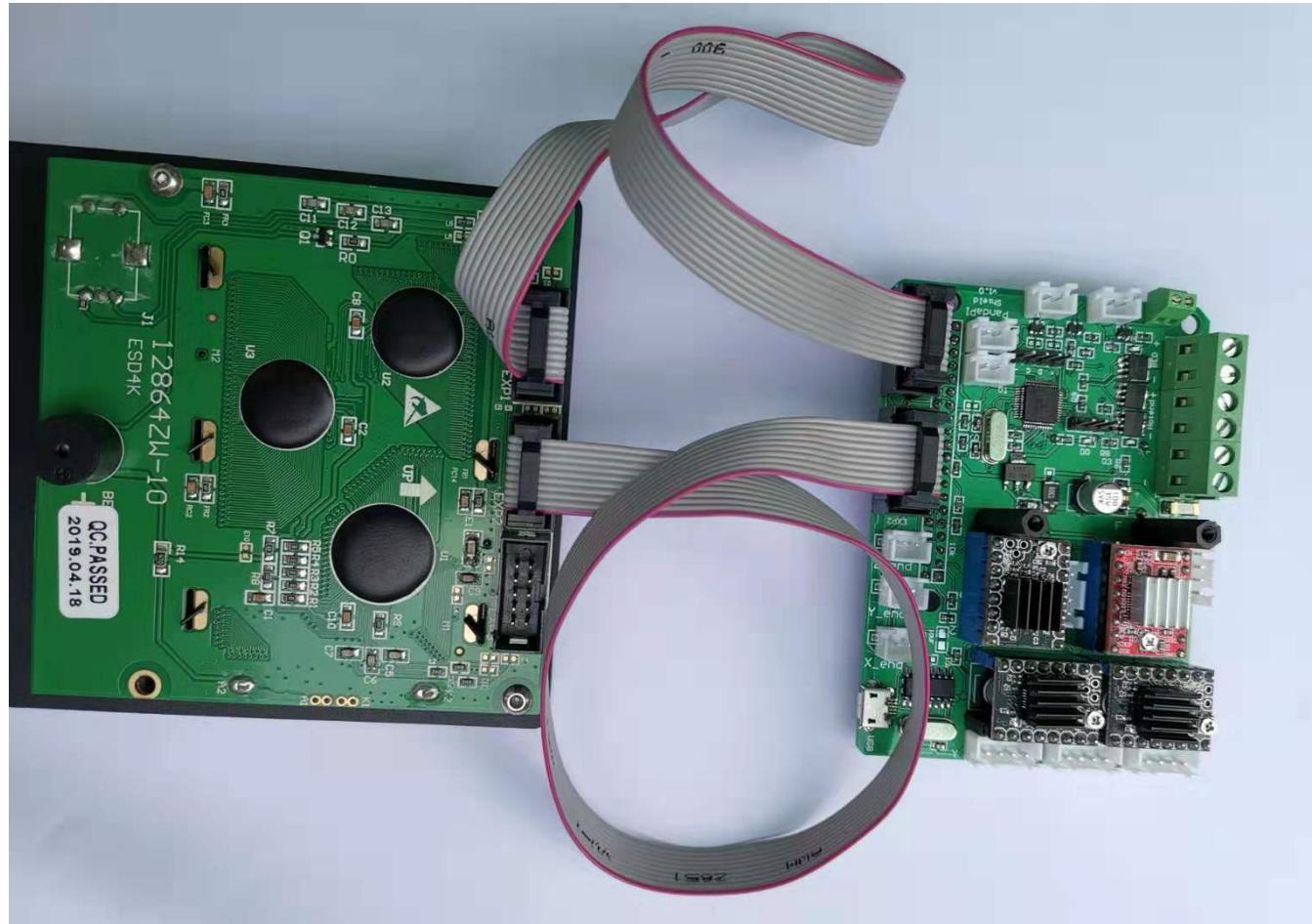
support 2 type of LCD128*64:

- CREALITY UniversalLCD128*64
- RepRapDiscount Full Graphic Smart Controller

1)connect to Creality UniversalLCD128*64(ender3 CR-10):

exp1---->exp1

exp2---->exp2

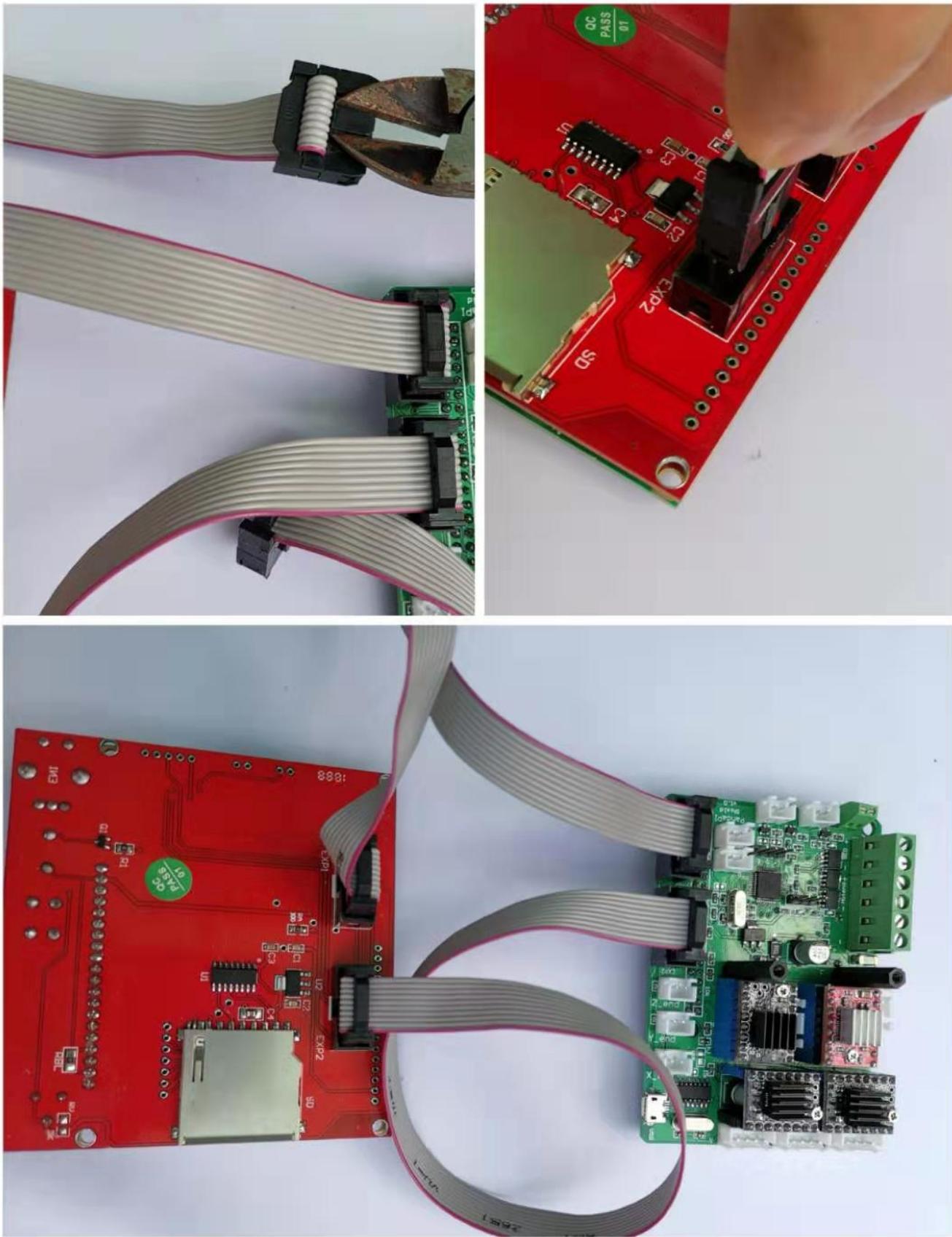


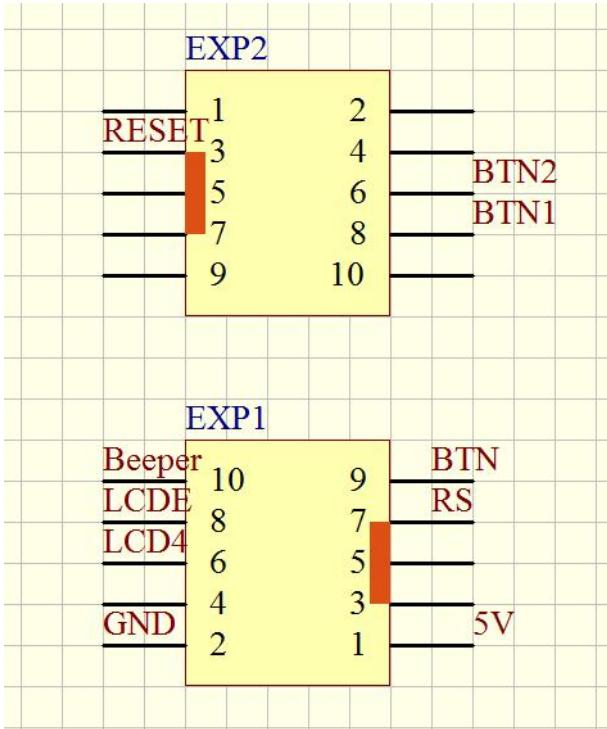
2)connect to RepRapDiscount Full Graphic Smart Controller

exp1---->exp1

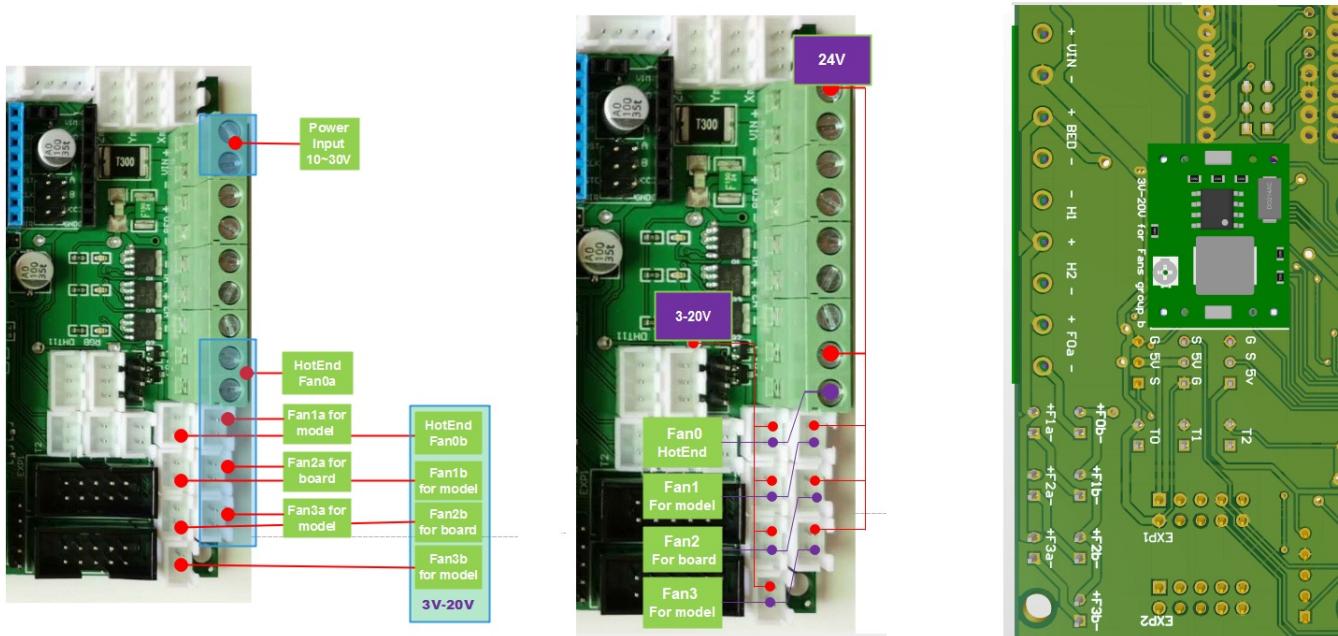
exp2---->exp2

The notch connectors on this LCD panel are reversed. Cut the notches on the cable and insert them upside down.





3. Fans:



name	24V	3V-20v		gcode control
FAN0	Fan0a	Fan0b	hotend	it will be opened automatically if the temperature of Hotend0 is higher than 60°C.
FAN1	Fan1a	Fan1b	model cooler	M106: Set Fan Speed 0~255, E.g "M106 P0 S255"
FAN2	Fan2a	Fan2b	for motor driver	it will be opened automatically if the driver is working.
FAN3	Fan3a	Fan3b	model cooler	M106: Set Fan Speed 0~255,M107: Fan Off. E.g "M106 P1 S255"

Because the MCU control all the Fans and heaters, and it can not be changed by the marlin code. if you want to re-define them the best way is to re-splice the fan connectors.

How-to-Flash-img-&-WIFI-setup

1. RaspberryPi img of PandaPi download link:

[PandaPi_3/4 img google drive](#)

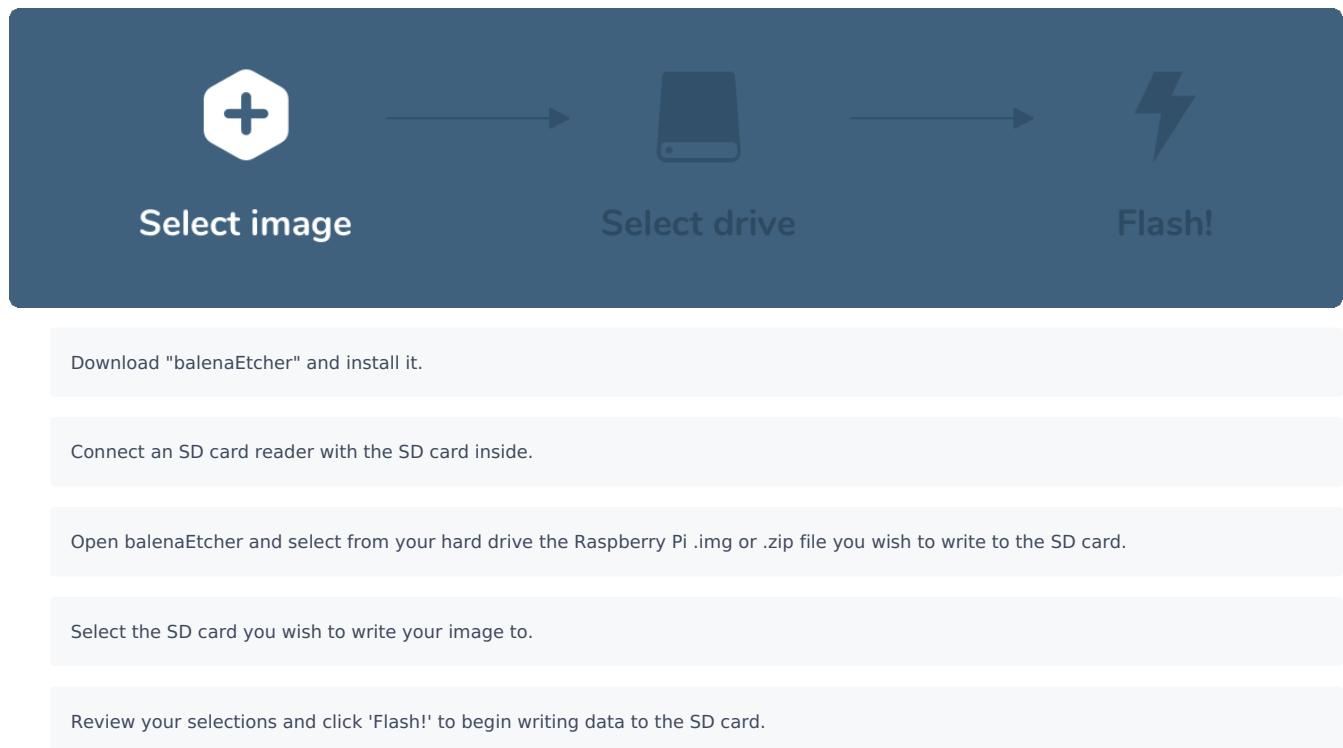
[PandaPi_3/4 img MEAG drive](#)

2. Writing this image to the SD card

Before you start, don't forget to check the SD card requirements.

You will need to use an image writing tool to install the image you have downloaded on your SD card.

balenaEtcher is a graphical SD card writing tool that works on Mac OS, Linux and Windows, and is the easiest option for most users. balenaEtcher also supports writing images directly from the zip file, without any unzipping required. To write your image with balenaEtcher:



3. WiFi setup

Configure your WiFi connection by editing `octopi-wpa-supplicant.txt` on the root of the flashed card when using it like a thumb drive. Important: Do not use WordPad (Windows) orTextEdit (MacOS X) for this, those editors are known to mangle the file, making configuration fail. Use something like Notepad++, Atom or VSCode instead or at the very least heed the warnings in the file. For most WiFi networks, you'll edit the section that looks like this:

```
## WPA/WPA2 secured
network={
    ssid="JoesWiFi"
    psk="12345"
}
```

enter your WiFi network `ssid` (your network's "name") and the password in the indicated places. Note that the `ssid` and password are case sensitive. (A common problem when cutting and pasting this information is accidentally including leading or trailing spaces which are not part of the SSID or password.) For a network named `JoesWiFi` with a password of `12345`, it should look like one above.

4. Default password

the default login for SSH, Octopi or Share folder is `pi` with password `raspberry`

5. Shared folder of Pi (only 2021 later img)

To connect to your Shared folder on PI from Windows, begin by opening up the “File Explorer”.

`\\\192.168.xx.xxx\`

the marlin source code folder `\\\192.168.xx.xxx\pi\PandaPI\Marlin2.x\pandapi`

How-to-Find-Raspberry-IP

How to Find Raspberry Pi IP Address

method 1:Using the phone

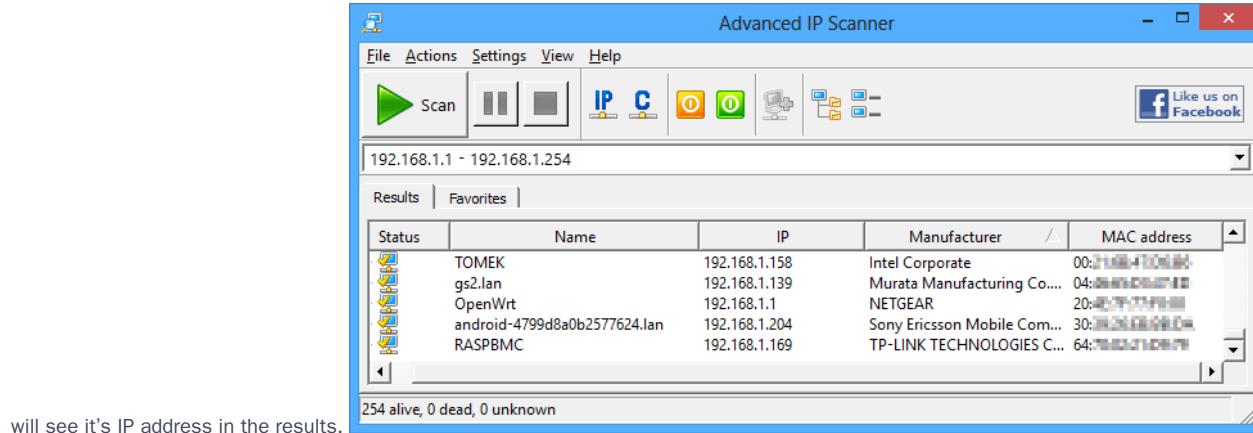
Fing

method 2:Using the laptop or desktop PC

IP Scan

You can easily find the IP address of your Raspberry Pi using the [IP Scan](#) on your entire network. You may know that almost all of the home networks contain the combination of 254 IP addresses. So in the end, it's an easy task to find Raspberry Pi IP address.

There are some third party software available on the web which scans the entire network, and if you have connected your Pi to the system, then you



will see it's IP address in the results.

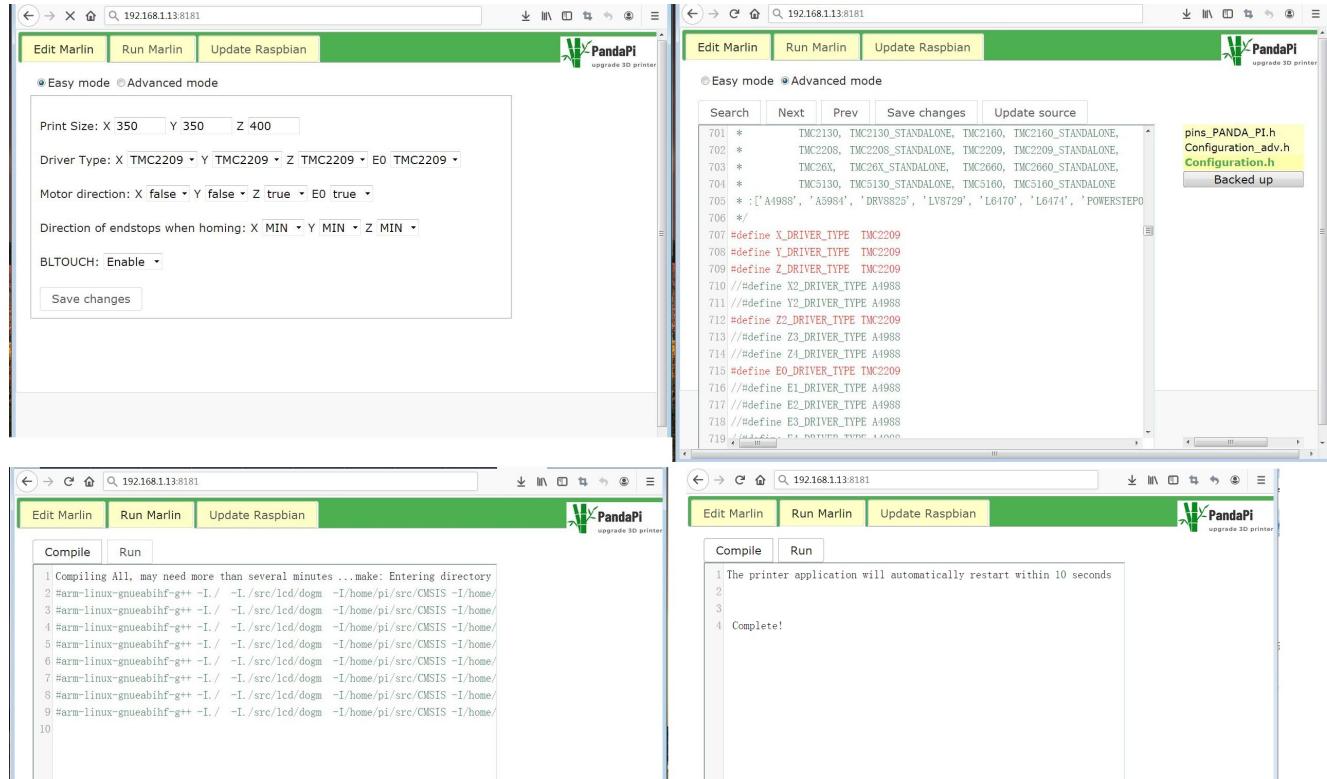
How-to-configure-Marlin-code

the default configure of Marlin in the Pi IMG is for printer ender3,for other printers you may need to configure it.

There are two ways to compile marlin:

Edit and compile with Browser

1. open the http://RaspberryPi_IP:8181/ with browser on your computer or phone. here my PI's IP address is 192.168.1.13



2. Save changes. If you changed the source code e.g Configuration.h, you need to save it.

3. Compile. the raspberryPi will compile the souce code and return status information that will be displayed on a console dialog.

4. Run. if compile success, you can click this button to run(no need to reboot the raspberryPi).

Update Raspbian. Update this web edit tool or other files in the Raspberry Pi system.

Update source. Update to the newest marlin source code from pandapi github, this update will overwirte the local files, and the old files will be backed up.

this PandPi source code based on marlin firmware ,except the temperature and fan control code that are running on pandaPI MCU, so you can reference <http://marlinfw.org/docs/configuration/configuration.html>

Compile it with command line(SSH)

setup ssh <https://www.raspberrypi.org/documentation/remote-access/ssh/windows.md>

1. compile source code:

```
cd /home/pi/PandaPI/Marlin2.x/pandapi
```

```
ls
```

you can see all the marlin source code are here.

"/home/pi/PandaPI/Marlin2.x/pandapi" is the default source code folder, you can copy you source code to any folder you like.

then run "make" to compile

```
sudo make clean;sudo make
```

by the way,you can download the source code from github,and copy the pandapi folder to the U drive,plug it to raspberry Pi and then run make command in the pandapi folder.

2. kill pi_marlin:

```
sudo killall monitor.sh
```

```
sudo killall pi_marlin
```

password of sudo is raspberry

3. copy the pi_marlin to boot up folder

```
sudo chmod 777 pi_marlin
```

```
cp pi_marlin /home/pi
```

4. run marlin

```
sudo ./pi_marlin &
```

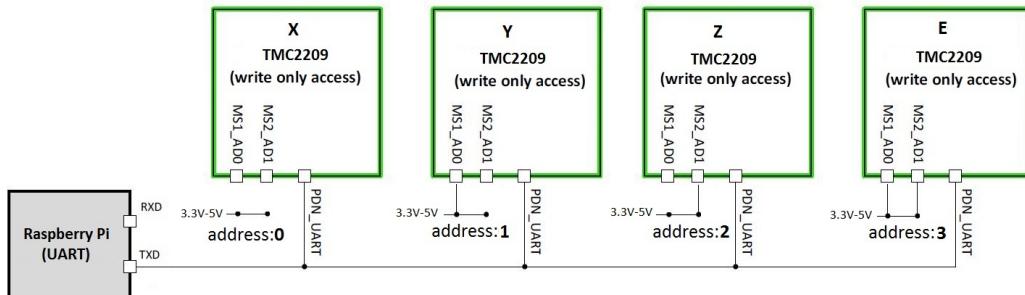
How-to-run-TMC2209-with-UART-mode-(V2.0-2.5)

- **Wiring**

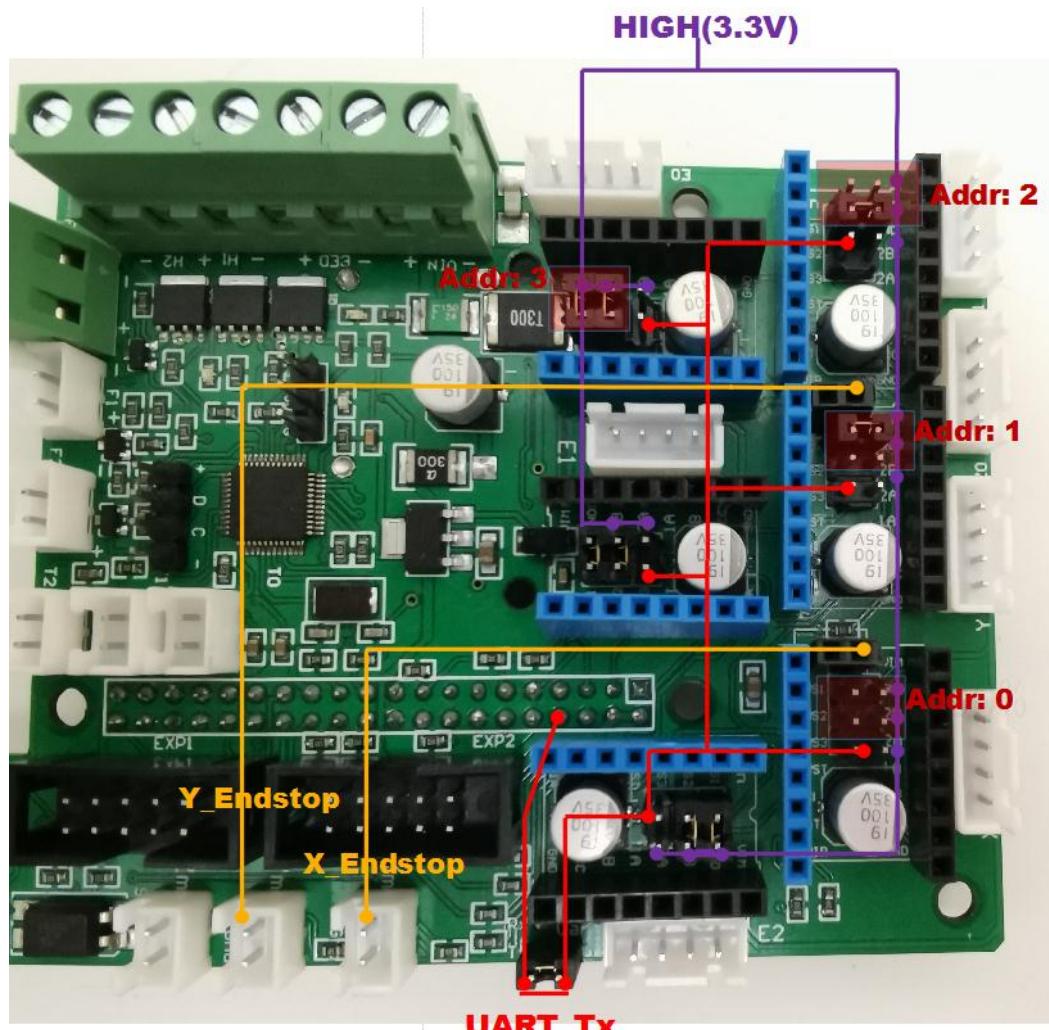
BIGTREETECH TMC2209 driver

TMC2209 datasheet

- Attaching the TMC2209 to raspberrypi UART



- For Pandapi V2.0



- For Pandapi V2.5

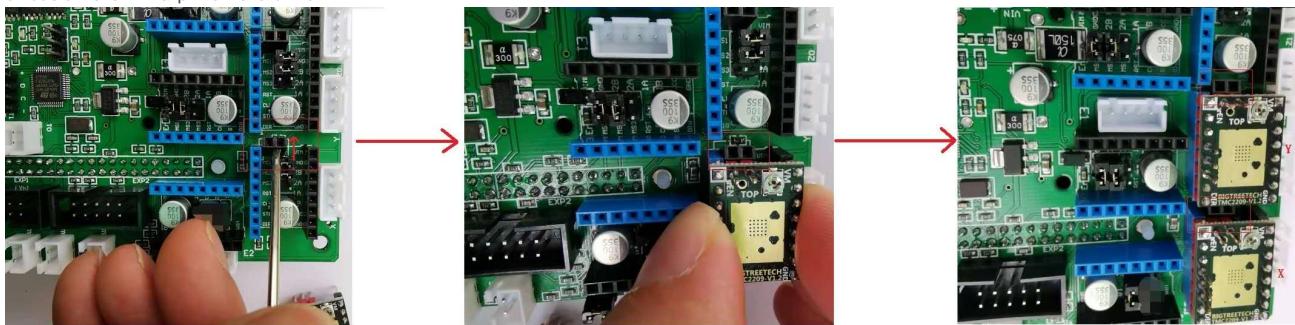


- Jumpers setting for uart and address of TMC2209, and then plug in TMC2209

1. For sensorless homing, you must plug in the DIAG pin which is connected to the limit endstop pin directly in PCB.



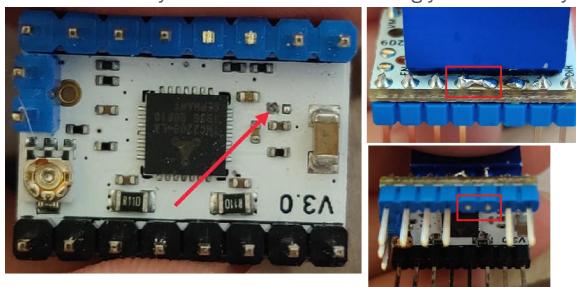
2. If you use limit endstop instead of sensorless homing, the DIAG pin is not used, and not allowed to plug in, so you can bend it like the following or cut off the DIAG pin on the driver.



Disable Sensorless Home

If you use X/Y Endstop, please disable sensorless home

3. If you use TMC2209 v3.0 from Fysect. To make UART working you must mod your driver. Cut one wire and remove two jumpers. Then solder. Take a look at pictures



a look at pictures

- Configure TMC2209 for Marlin2.0.x

1. the address setting is in the Configuration_adv.h

```
/**  
 * Four TMC2209 drivers can use the same HW/SW serial port with hardware configured addresses.  
 * Set the address using jumpers on pins MS1 and MS2.
```

```
* Address | MS1 | MS2
*   0 | LOW | LOW
*   1 | HIGH | LOW
*   2 | LOW | HIGH
*   3 | HIGH | HIGH
*
* Set *_SERIAL_TX_PIN and *_SERIAL_RX_PIN to match for all drivers
* on the same serial port, either here or in your board's pins file.
*/
#define X_SLAVE_ADDRESS 0
#define Y_SLAVE_ADDRESS 1
#define Z_SLAVE_ADDRESS 2
#define E0_SLAVE_ADDRESS 3
#define E1_SLAVE_ADDRESS 3
#define E2_SLAVE_ADDRESS 3
```

2. Edit the "Configuration.h", open the sensorless homing and X/Y driver type to TMC2209.

before:

```
#define X_DRIVER_TYPE A4988
#define Y_DRIVER_TYPE A4988
```

after:

```
#define X_DRIVER_TYPE TMC2209
#define Y_DRIVER_TYPE TMC2209
```

3. make sure the baudrate to 115200

```
#define BAUDRATE 115200
```

4. Open the sensorless homing in Configuration_adv.h

before:

```
//#define SENSORLESS_HOMING // StallGuard capable drivers only
```

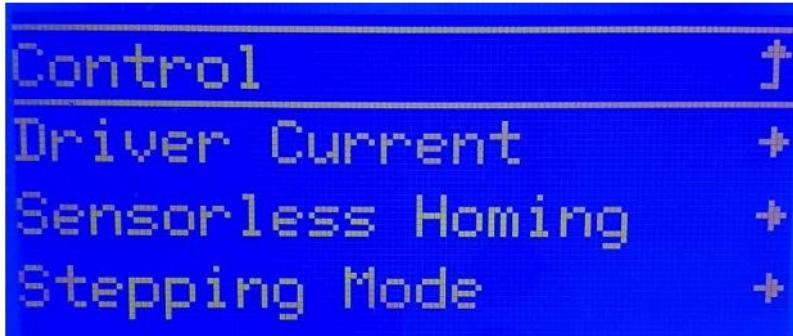
after:

```
#define SENSORLESS_HOMING // StallGuard capable drivers only
```

click"Save this file" and "Verify", and then "Run"

you can also set the current or sensitive value of the sensorless homing on the screen.

Info Screen-->Configuration-->Advanced Settings-->TMC drivers



FAQ :

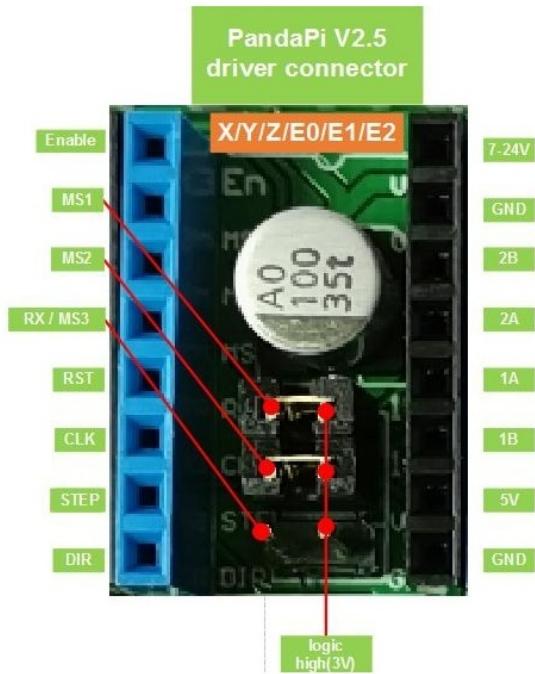
```
Testing X connection... Error: All LOW  
Recv: Testing Y connection... Error: All LOW  
Recv: Testing Z connection... Error: All LOW  
Recv: Testing E connection... Error: All LOW
```

The reason of this TMC Error message is that with write only access to TMC, and there is no RXD connection in the hardware. So that's normal.

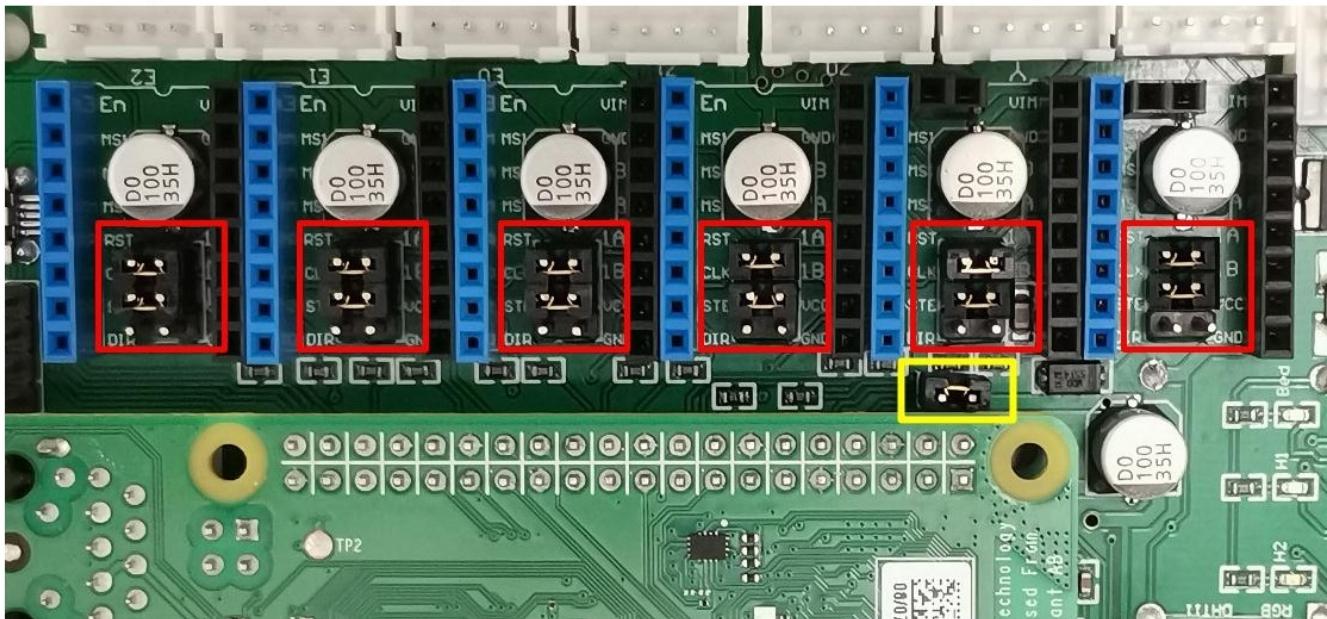
Have fun!

[sensorless homing video](#)

Run-TMC2208-with-UART-mode



Jumpers for using TMC2208 as UART mode



- Jumpers setting for uart and address of TMC2208

In TMC2208 UART mode, plugin jumpers on MS1/MS2, and keep MS3 no jumper.

- **Configure TMC2208 for Marlin2.0.x**

2. [Edit](#) the "Configuration.h", enable driver type to TMC2208.

before:

```
#define X_DRIVER_TYPE A4988  
#define Y_DRIVER_TYPE A4988  
#define Z_DRIVER_TYPE A4988  
#define E0_DRIVER_TYPE A4988
```

after:

```
#define X_DRIVER_TYPE TMC2208  
#define Y_DRIVER_TYPE TMC2208  
#define Z_DRIVER_TYPE TMC2208  
#define E0_DRIVER_TYPE TMC2208
```

3. make sure the baudrate to 115200

```
#define BAUDRATE 115200
```

click"Save this file" and "Verify", and then "Run"

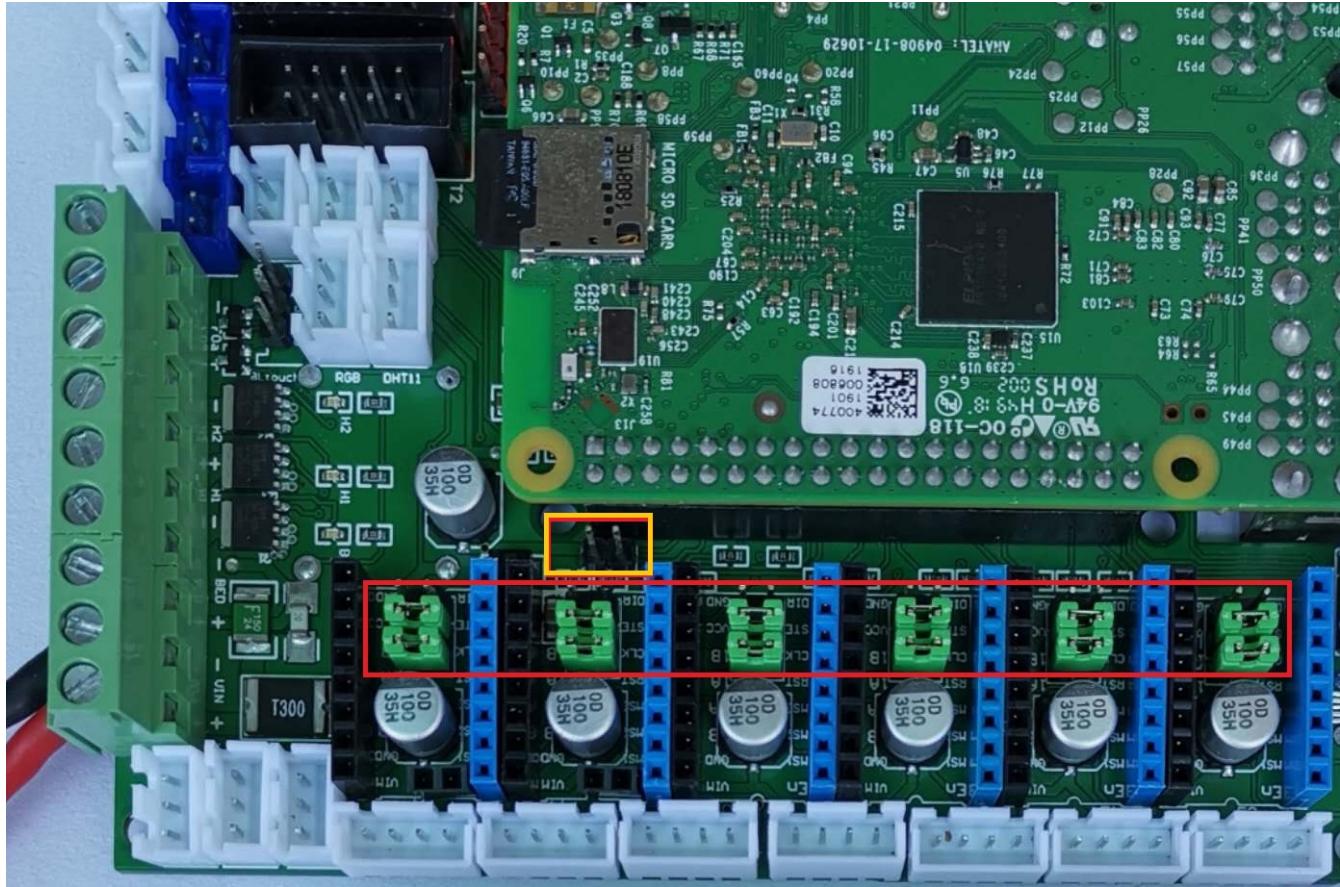
note: there is only one tx line and no address for tmc2208,so we can not set the motors current in different value.that's means the current value of all the motors are setted in the same value as the last motor's value, here the last motor is E0.

Have fun!

How-to-run-TMC2208-TMC2209-with-standalone-mode

Standalone mode for TMC2208/2209

1. We need to plug the jumpers on the MS1 and MS2 for 1/16 microstep, as shown in the red box.
2. Disconnect the UART control TX pin by unplugging out the jumper on the TX pin as shown in the yellow box.



in Standalone mode, you can set the current by adjusting the voltage with the potentiometer.

Have fun!

How-to-setup-your-own-configuration

**WARNING !!! - If you do this you won't get updates from
<https://github.com/markniu/PandaPi> anymore and have to merge
changes manually**

To setup your own machine you need to fork this repo and create a branch.

As an example if you want to use the PandaPi board for a Prusa MK3 you can create the branch `MK3`

To switch to that branch on your Pi you have to login to it via ssh.

Then do:

```
cd ~  
mv PandaPi PandaPi.old  
svn checkout https://github.com/<USER>/PandaPi.git/branches/MK3
```

You have to replace `<USER>` with your github username.

To be sure go to `<IP OF YOUR PI>:8181` in your browser and click on `update source` in the `Edit Marlin` tab when activated the Advanced mode.

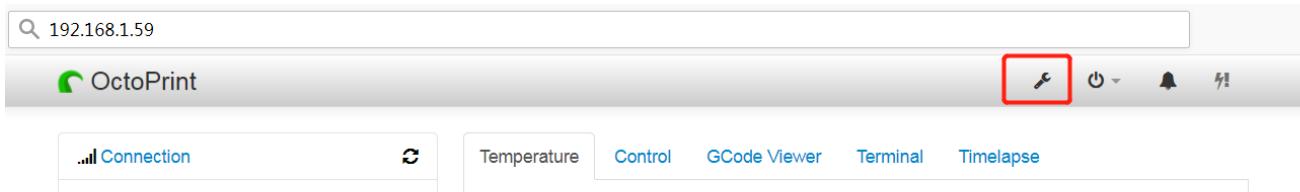
How-to-run-OctoPrint

Just need to change the 'serial settings' at the first time.

other settings are the same as normal.

how to change the serial setting and run:

1, Click Settings



2, add "/dev/tnt0" in the Additional serial ports and then save

The screenshot shows the OctoPrint Settings interface. The left sidebar has sections for PRINTER (Printer Profiles, Temperatures, Terminal Filters, GCODE Scripts), FEATURES (Features, Webcam & Timelapse, GCODE Visualizer, API, Application Keys), and OCTOPRINT (Server, Folders, Appearance, Logging, Plugin Manager, Software Update). The 'Serial Connection' tab is active. The 'General' tab is selected in the top navigation bar. In the 'Connection' section, the 'Serial Port' dropdown is set to 'AUTO' and the 'Baudrate' dropdown is set to 'AUTO'. There is an unchecked checkbox for 'Auto-connect to printer on server start'. Below that, in the 'Additional serial ports' section, the value '/dev/tnt0' is entered into a text input field. A note below the input field says: 'Use this to define additional glob patterns matching serial ports to list for connecting against, e.g. /dev/ttyAMA*. One entry per line.' At the bottom right are 'Close' and 'Save' buttons.

3, Choose the port and baud rate like bellow and then connect



Serial Port

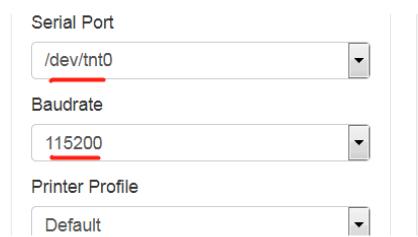
/dev/tnt0

Baudrate

115200

Printer Profile

Default

A screenshot of the OctoPrint configuration interface. It shows three dropdown menus: 'Serial Port' set to '/dev/tnt0', 'Baudrate' set to '115200', and 'Printer Profile' set to 'Default'. The background is white with black text and grey dropdown arrows.

How-to-plugin-OctoPrint-Enclosure(DTH11-temperature-humidity-sensor)

Wiring:



Example:

OctoPrint

Printer Safety Warning

Warning!
Your printer's firmware is known to lack mandatory safety features (e.g. thermal runaway protection). This is a fire risk.
[Learn more...](#)

Connection

State
State: **Operational**

File:
Timelapse: -
Approx. Total Print Time: -

Print Time: -
Print Time Left: -
Printed: -

Print **Pause** **Cancel**

Files

javascript:void(0)

Temperature Control GCode Viewer

Input 1
18 °C
85 %

300°C
250°C
200°C
150°C
100°C
50°C

Actual T: 17.0°C Target T: off

Actual Bed: 17.0°C Target Bed: off

	Actual	Target			Offset
Tool	17.0°C	-	off	°C	+ ✓ -
Bed	17.0°C	-	off	°C	+ ✓ -

reference link:

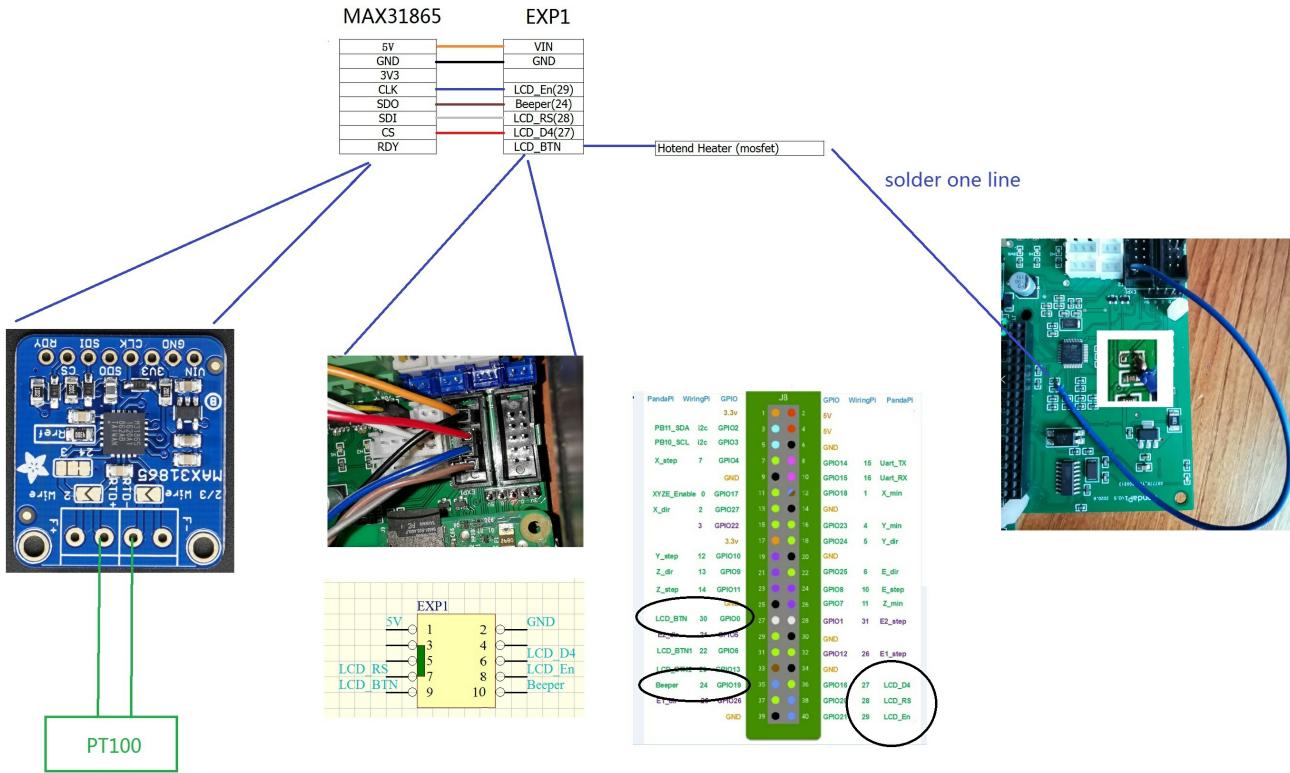
<https://plugins.octoprint.org/plugins/enclosure/?fbclid=IwAR0jliQRM-qFwlipZBYKKQwSMhEjacn2ydTK2Qq8PZgWSAI25TW1bRCDcxo>

<https://github.com/vitormhenrique/OctoPrint-Enclosure>

How-to-connect-to-PT100

1. connection:

Just connect 6 lines between the MAX31865 module and the EXP1 connector, and remove one resistor then solder 1 line to the pandapi board which is the control signal line of mosfet for the hotend0.also remember to unplug the old temperature sensor on the TO connector.



2. enable the max31856 in configuration.h

```
#define MAX31856_PANDAPI
```

if you want to use other GPIO for the max31856,you can edit the pins in file pins_PANDA_PI.h.

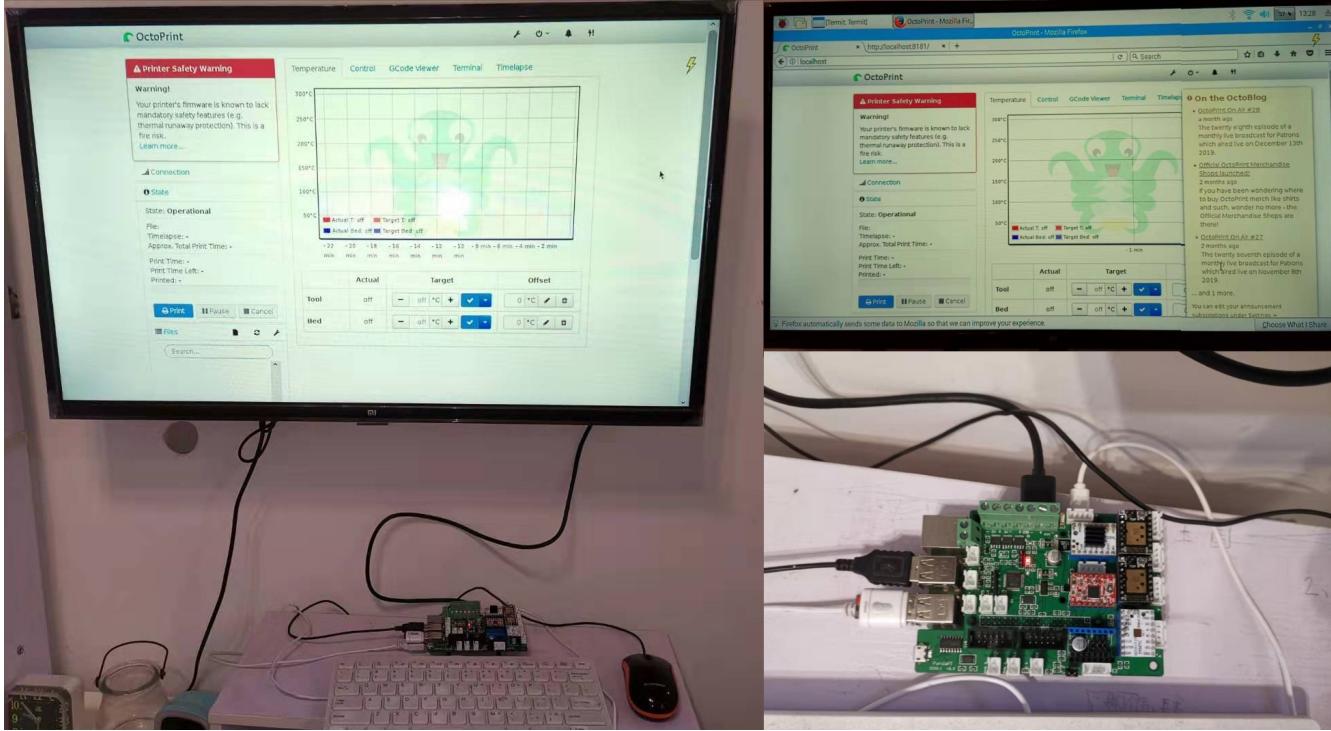
testing video:<https://youtu.be/LhGSKaXKsbw>

note: please keep one eye on the hotend before you have test it ok and stable.

How-to-install--HDMI-LCD

Here are 2 way about how to install HDMI LCD screen:

1. Desktop



1. install desktop on raspberry pi

<https://octoprint.org/blog/2016/05/02/new-octopi-release-0-13-0/>

The desktop subsystem is not included by default anymore since most people don't actually use it (OctoPi is meant to run as a headless network appliance), but in case you need it, we also put a little helper script on there which will allow you to quickly

install the desktop environment. Just SSH or keyboard into your OctoPi instance, execute

```
sudo ./scripts/install-desktop
```

and then follow the instructions.

2. install firefox

```
sudo apt install firefox-esr
```

3. open <http://localhost> in firefox web browser.

2.Octoscreen

if you do not like desktop, you can install Octoscreen: No need for desktop, web browser... Just install and set up... Way to go! Install:

```
wget https://github.com/Z-Bolt/OctoScreen/releases/download/v2.5.1/octoscreen_2.5-1_armhf.deb
sudo dpkg -i octoscreen_2.5-1_armhf.deb
```

Config:

```
edit this file: /etc/octoscreen/config.
set OCTOPRINT_HOST to http://localhost
set OCTOSCREEN_RESOLUTION to at least 800x480
```

More info in this link: <https://github.com/Z-Bolt/OctoScreen>

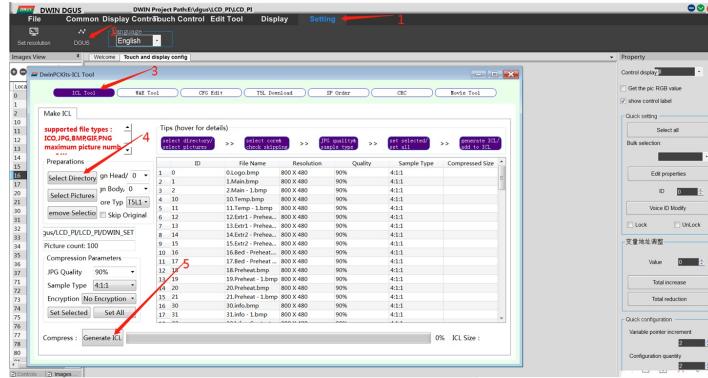
How-to-install-desktop(HDMI_LCD)

empty

Customize-the-UI-(replace-Image)

For example, if you want to replace the logo image, just replace the O.Logo.bmp(in the DWIN_SET) as your own. And then following the steps to generate and upload the ICL file. please download the LCD source file and tools here: https://github.com/markniu/Panda_LCD800480D

1. Run DgusToolV7.616.exe;
2. Setting-->DGUS ;
3. Select the ICL Tool-->Select Directory-->select the image directory e.g.DWIN_SET ;
4. Generate ICL--> Save as and replace the older "32.icl"



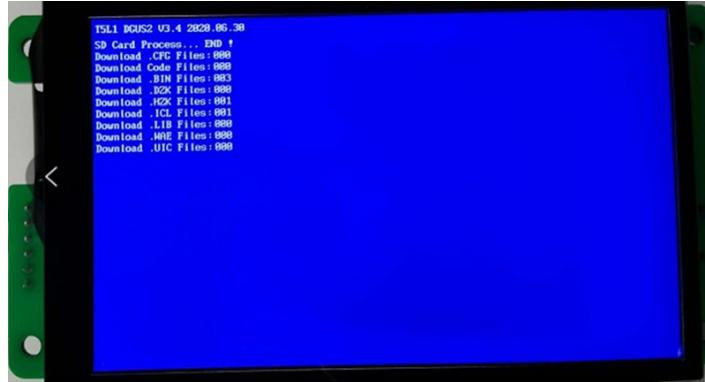
5. Upload screen Image to screen

- Format the TF card and copy files run windows command system as administrator, then enter :

```
format/q g:/fs:fat32/a:4096
```

Note: g is the drive letter of your TF card , and the card size must be 1-16G.

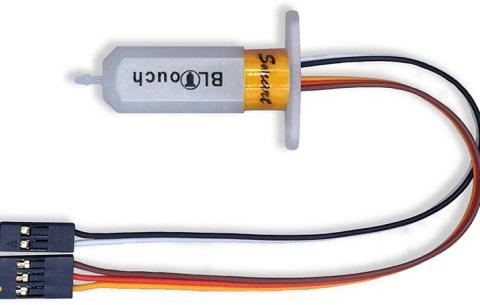
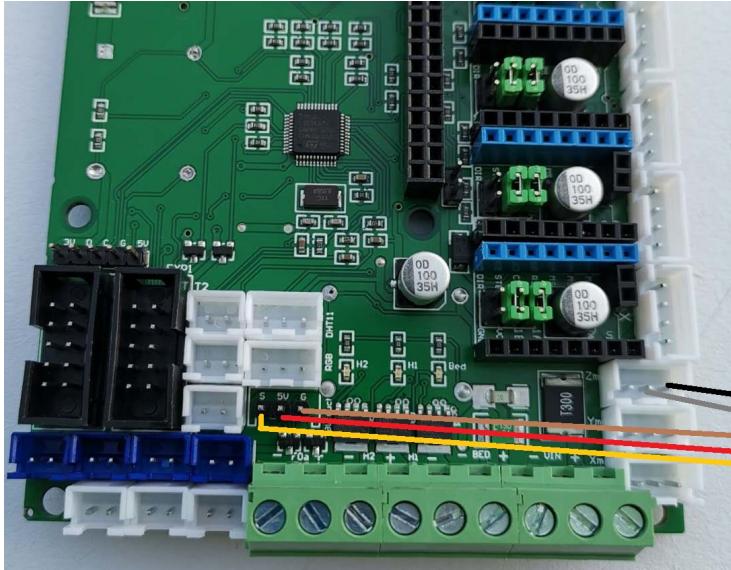
- Copy the folder DWIN_SET which contained your newest 32.icl to TF card;
- Insert the TF card into the card socket (back side of the screen) ,and then power on the screen. Wait for the blue screen and appear on the first line of the screen “SD Card Process... END !” it's may take 1-2minutes. Turn off the power and unplug out the TF card, re-power the screen Wait for the boot screen.



How-to-wire-BLtouch

Wiring

Pin	BLTOUCH	PandaPi
3Pin	Brown(GND) , Red(5V) , Orange(control signal)	5V , G(GND) , S(control signal)
2Pin	Black(GND) , White(Zmin)	G(GND) , Zmin



- BLTOUCH3.X:

Logic Voltage Free,no need to change anything.

Configure software

1. Edit the "configuration.h", find the BLtouch and enable it like the following

before:

```
// #define BLTOUCH
```

after:

```
#define BLTOUCH
```

Calibrating BLTouch from LCD display controls (Marlin1.x)

Assumptions:

The value of PROBE_OFFSET_FROM_EXTRUDER is set at -2 in Marlin configuration.h if it is set at something else take that into account in the formula below.

```
#define Z_PROBE_OFFSET_FROM_EXTRUDER -2 // Z offset: -below [the nozzle] (always negative!)
```

Control > Restore failsafe (start with default settings if first installation)

Prepare > Auto home

Prepare > Move axis > Move 0.1mm > Move Z

Slowly move the Z axis down until you have the correct first layer gap (paper or thin card method)

Note the distance on the display e.g. 0.6 mm (0.6 mm is example, note your actual)

Use this formula to determine your Z offset needed:

Z offset = -2 + 0.6

Z offset = -1.4mm

Control > Motion > Z offset and enter the value obtained above e.g. -1.4 in this example

Control > Store memory

Fine tuning – after initial setup.

The BLTouch is very accurate and consistent after the initial setup but there are times when you might want to fine tune for example for different filament materials or bed types.

Control > Motion > Z offset and enter the new value e.g. -1.3

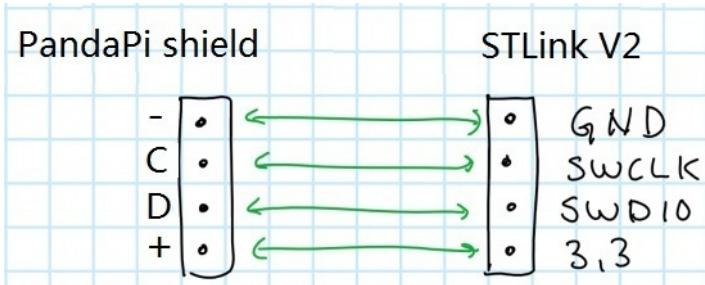
Control > Store memory

mount BLTOUCH download: <https://www.thingiverse.com/thing:3584158>

How-to-flash-MCU-firmware

Not recommend manual MCU firmware flash unless there is new firmware update.

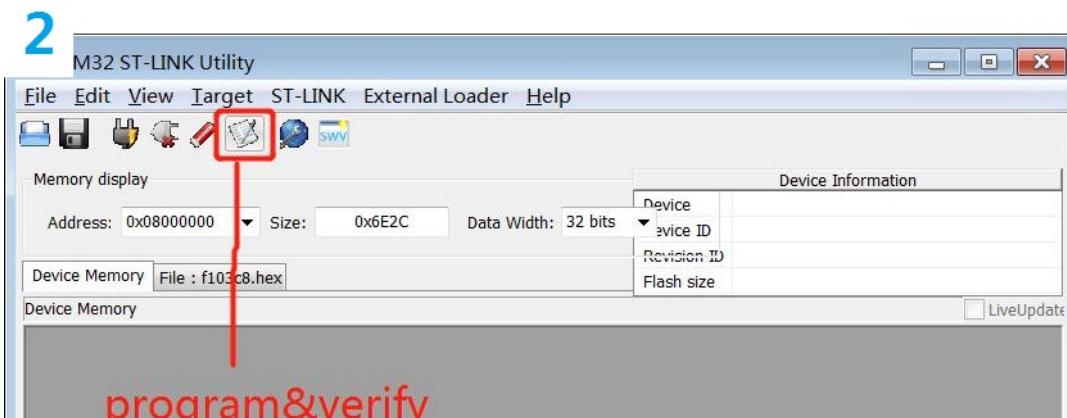
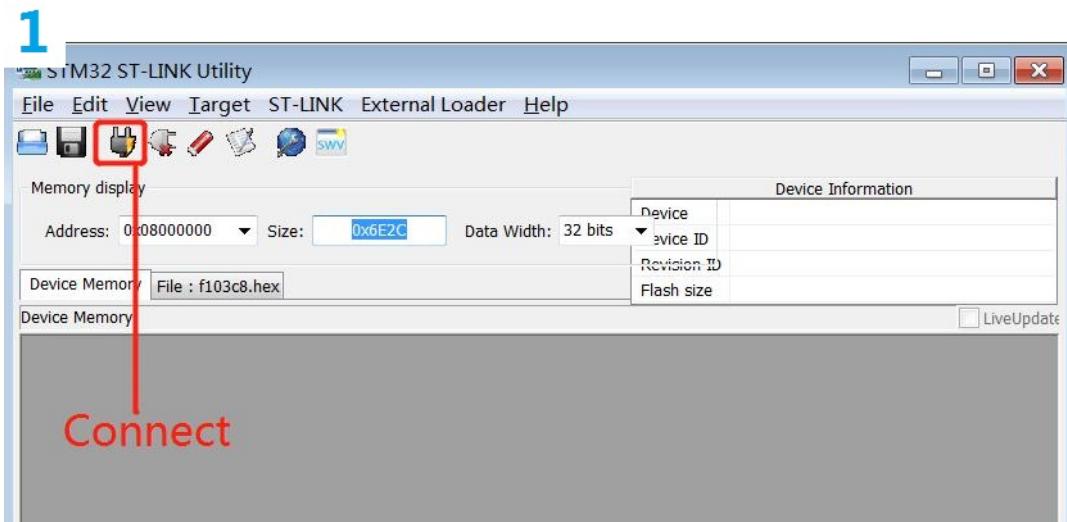
1. Connect stlinkV2(about \$3) to board.

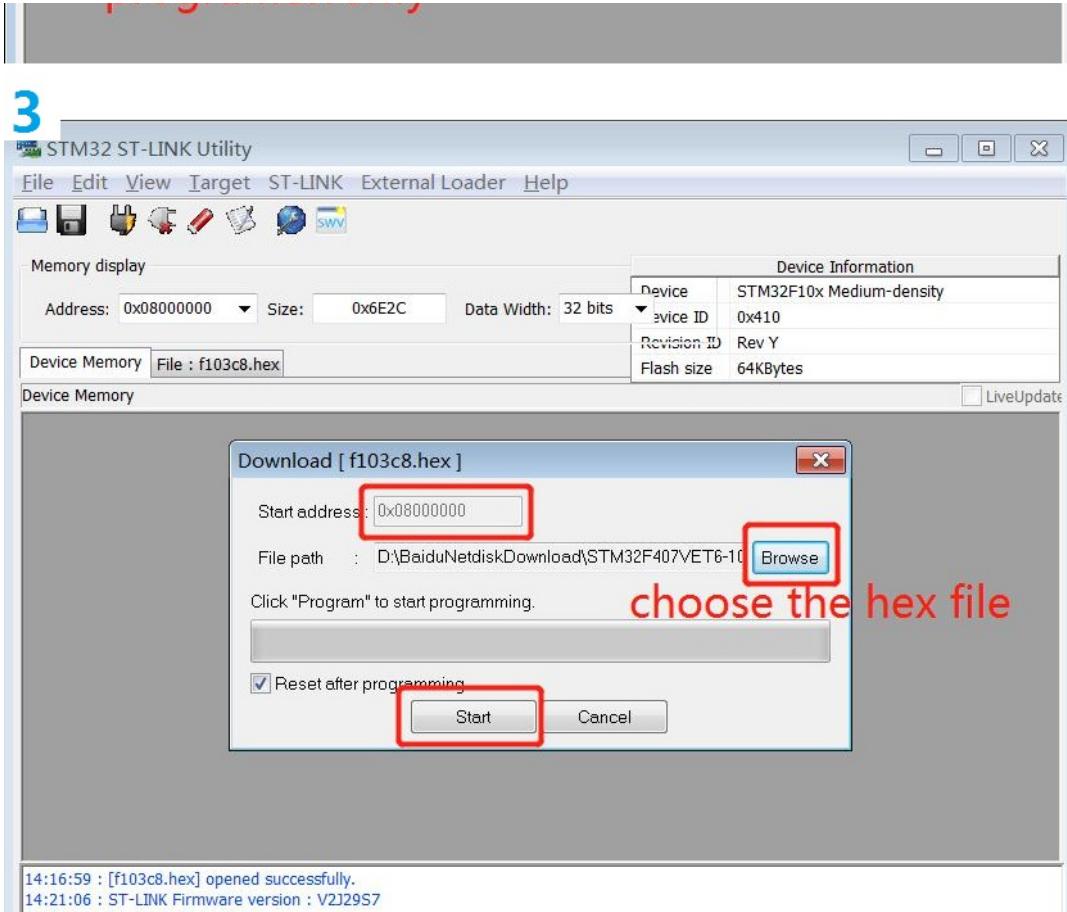


2. Install flash tool STM32 ST-LINK Utility

3. Download MCU firmware: [MCU hex file](#).

4. Connect STLinkV2 to PC, Start to flash.



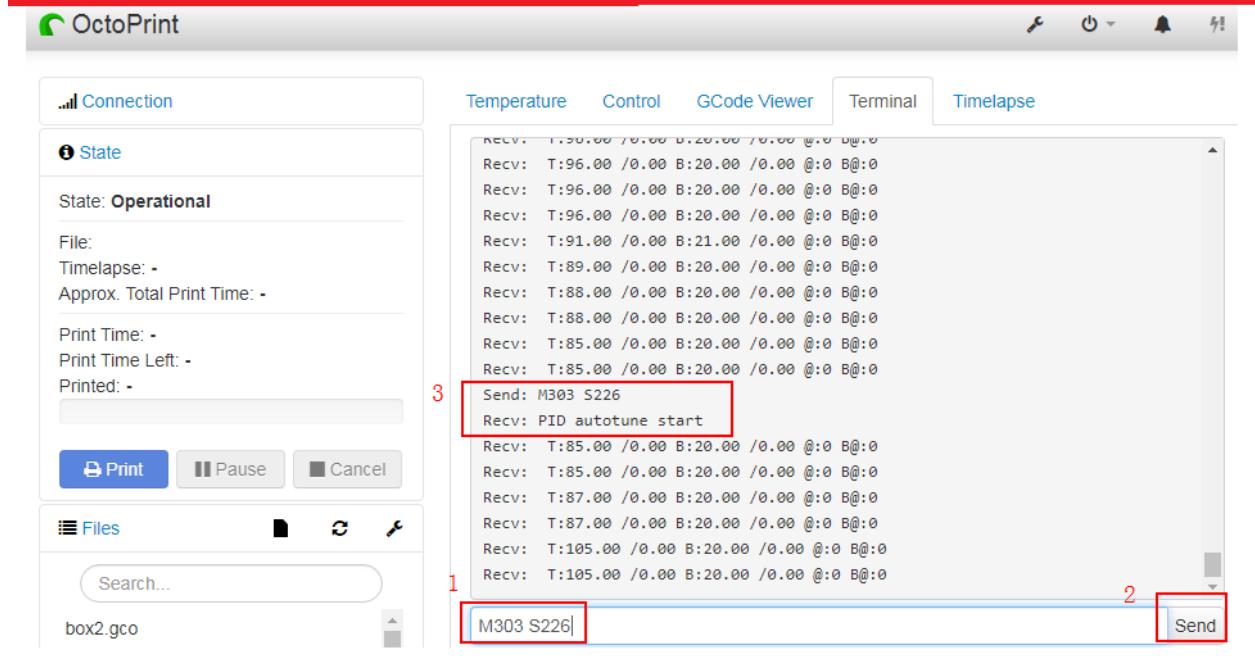


How-to-use-PID-auto-tune

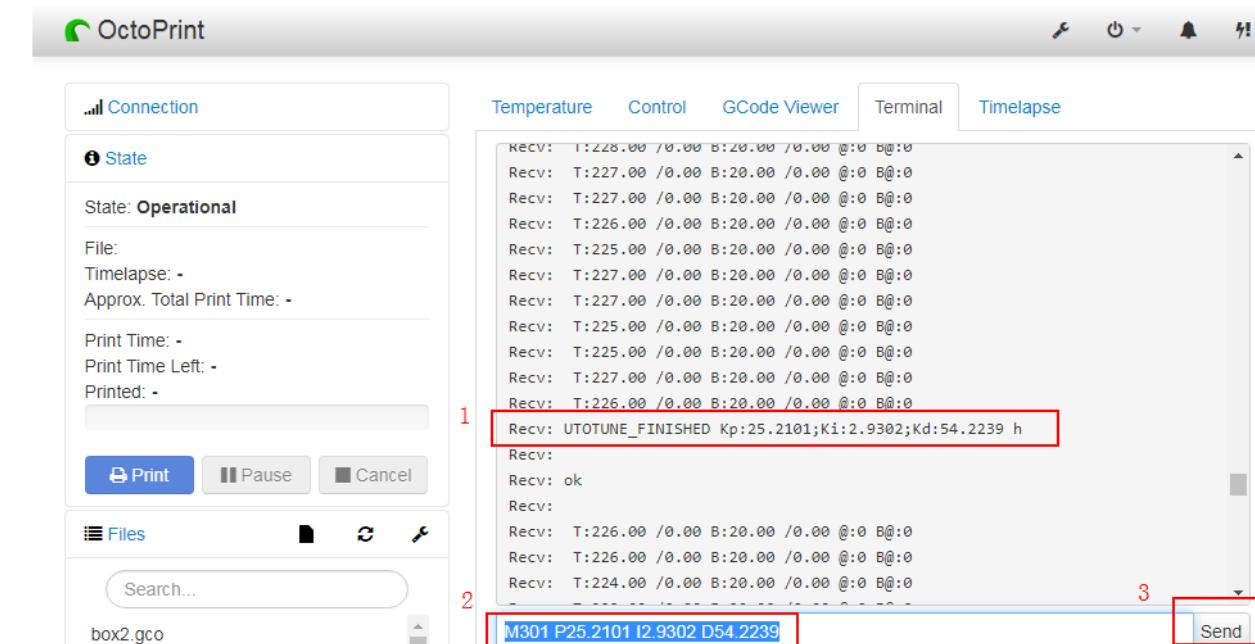
M303 - PID autotune

This command initiates a process of heating and cooling to determine the proper PID values for the specified hotend

Usage: M303 C<count> [E<index>] S<temperature> U<flag>
e.g. M303 S226



Waiting



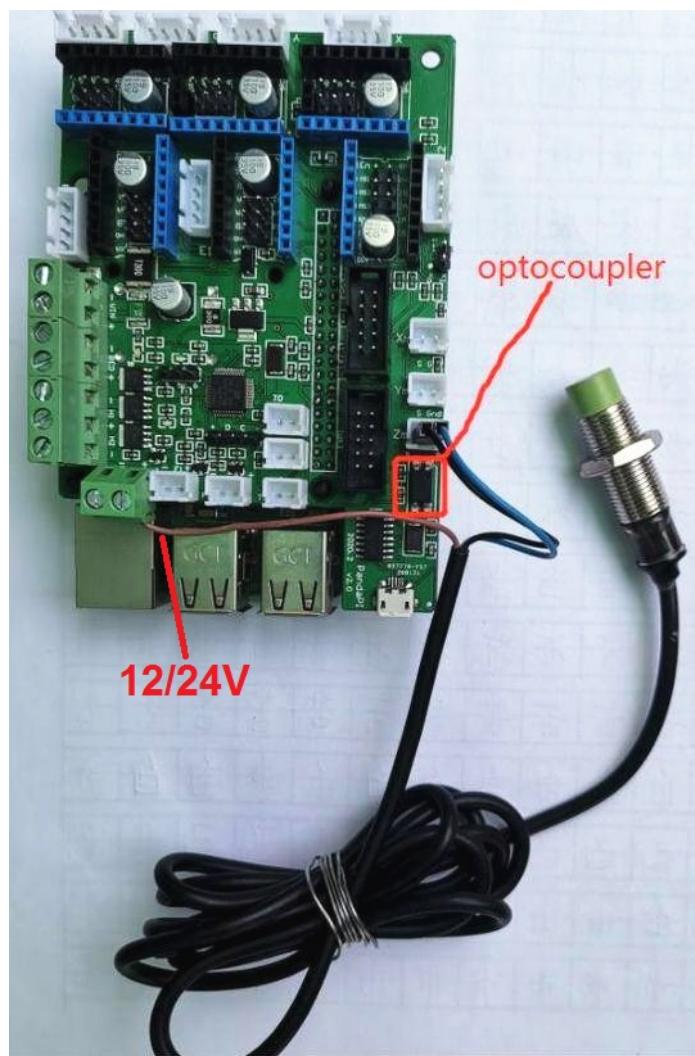
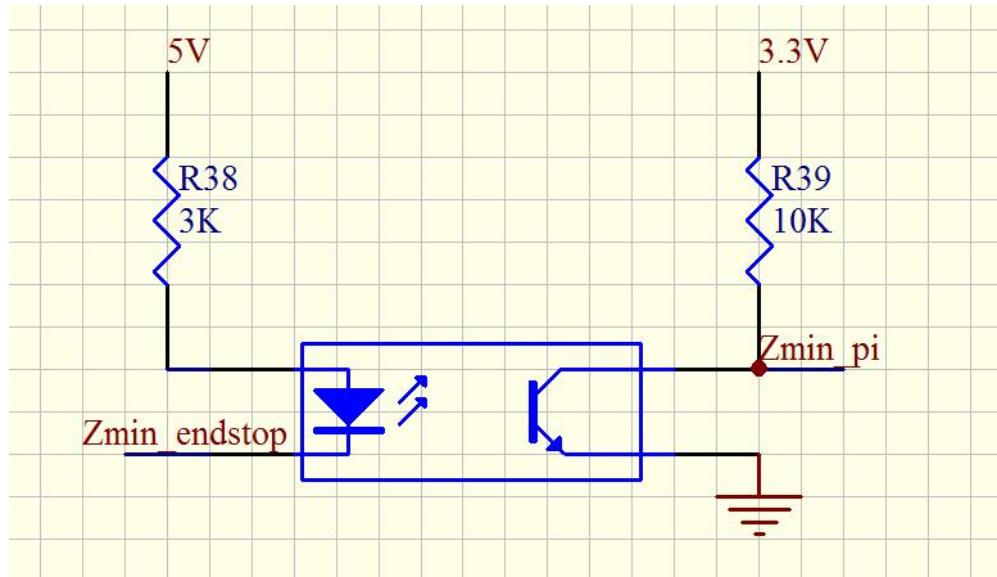
Set the PID value with gcode command M301

Store to memory with gcode command M500



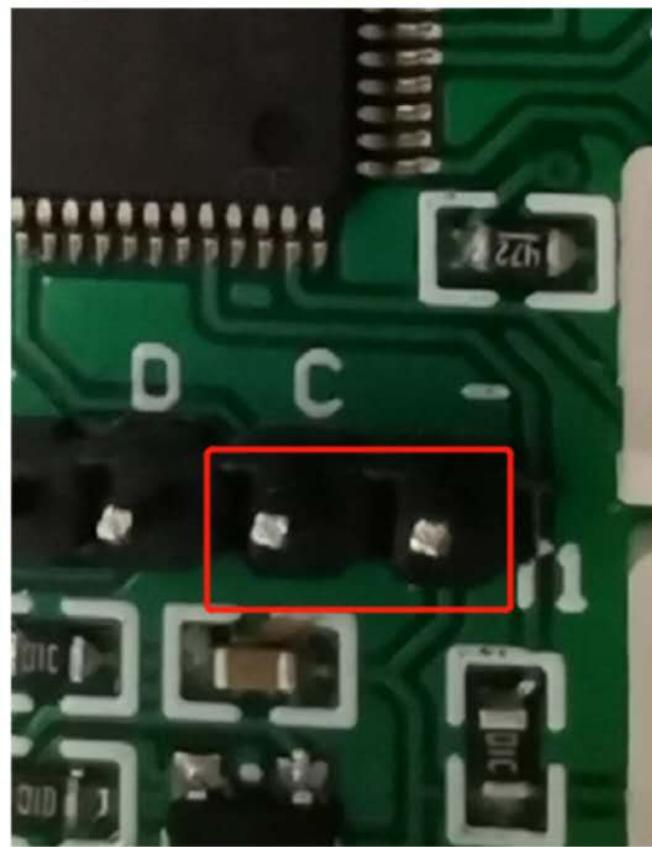
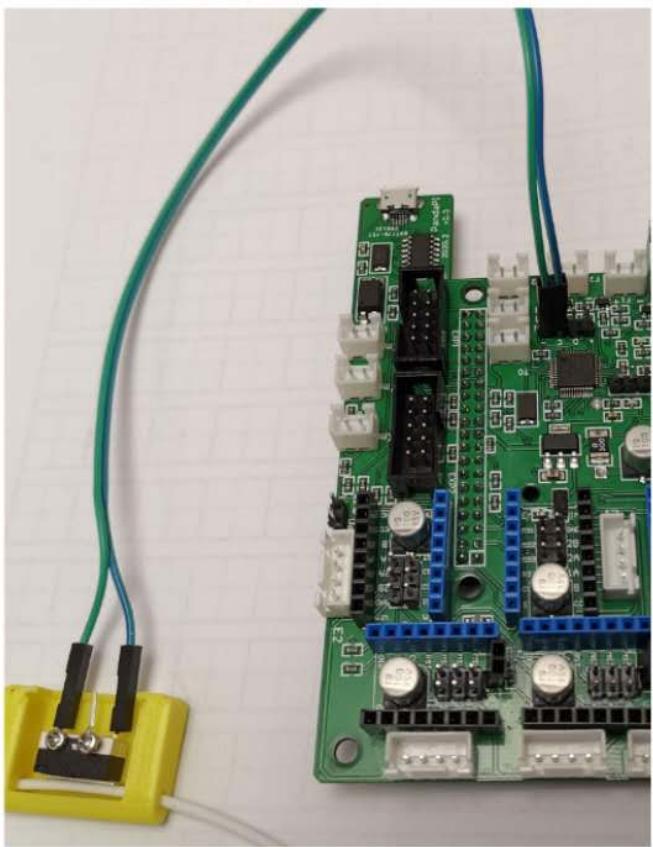
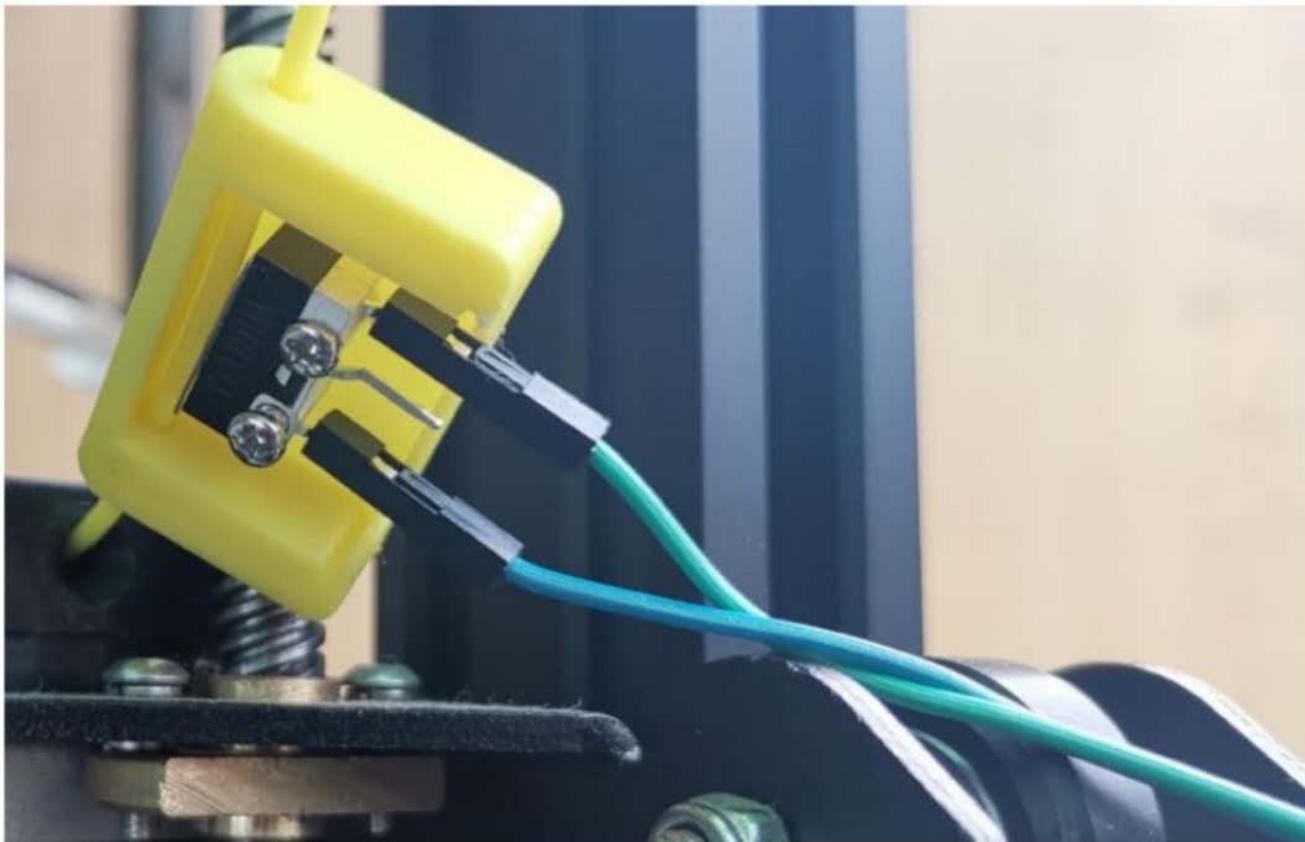
How-to-wire-Proximity-sensor

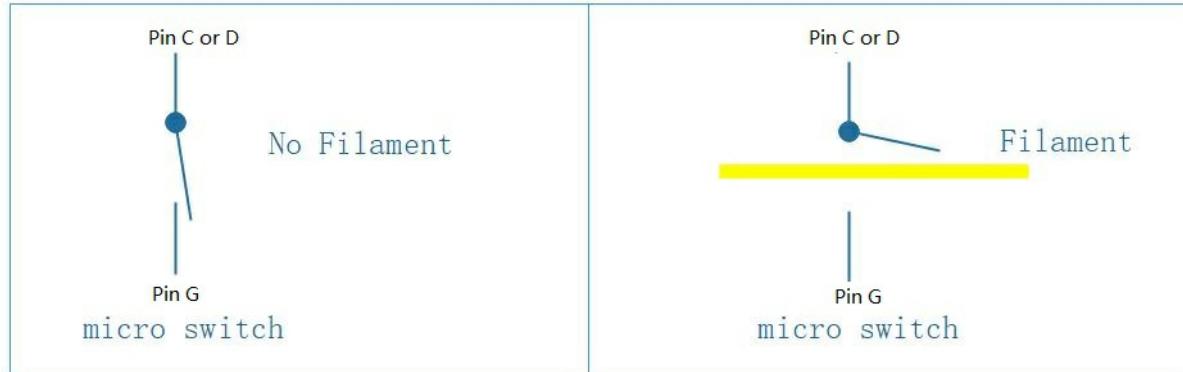
support wild range voltage 5V~36V Inductive sensor NPN, since it have optocoupler.



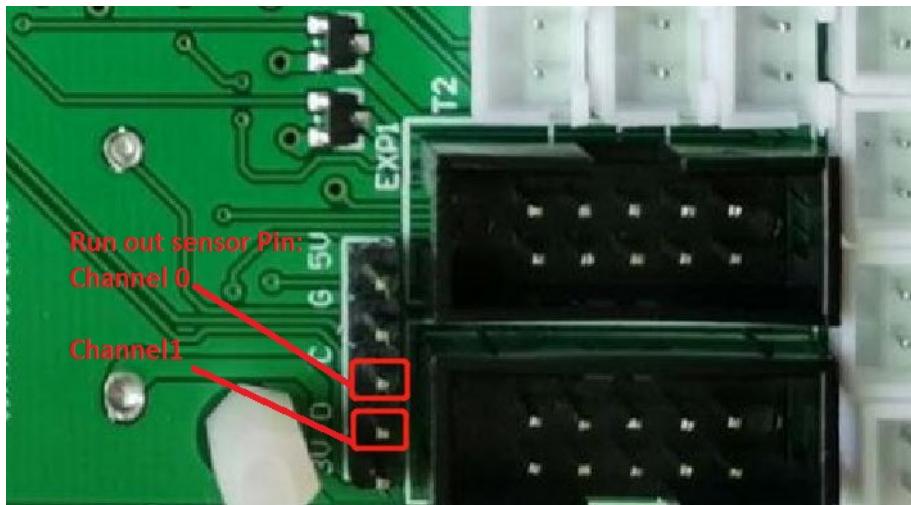
How-to-wiring-filament-runout-sensor

PandaPI V2.0





PandaPI V2.5



Run out sensor Pin:

Channel 0, Pin C (DEBUG_SCL PA14 in schematic)

Channel 1, Pin D (DEBUG_SDA PA13 in schematic)

Gnd, Pin G

make sure the "FILAMENT_RUNOUT_SENSOR" and "NOZZLE_PARK_FEATURE" are enabled in configuration.h

```
#define FILAMENT_RUNOUT_SENSOR
```

```
#define NOZZLE_PARK_FEATURE
```

make sure the "ADVANCED_PAUSE_FEATURE" is enabled in configuration_adv.h

```
#define ADVANCED_PAUSE_FEATURE
```

If you want to debug and find more, you can search "FILAMENT_RUNOUT_SENSOR" in the temperature.cpp and "poll_runout_pins" in the runout.h.

Remap-BED-Hotend0-Hotend1-(-heater_thermistor)

for example, if you want use the second hotend heater/thermistor as the Bed heater/bed thermistor,

just change the index number in pins_PANDA_PI.h as follow.

before:

```
#define HOTBED_CODE    0
#define HOTEND_0_CODE   1
#define HOTEND_1_CODE   2
```

after:

```
#define HOTBED_CODE    2
#define HOTEND_0_CODE   1
#define HOTEND_1_CODE   0
```

upgrade-to-printer-ender5

Since the default configuration is for Ender3, the following 4 points of ender5 should be changed.

1. the PID for hotend

Ender3:

```
#define DEFAULT_Ki 2.0
```

Ender5:

```
#define DEFAULT_Ki 1.08
```

2. Axis Steps for Z Ender3:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 }
```

Ender5:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 800, 93 }
```

3. Stepper direction Ender3:

```
#define INVERT_X_DIR true  
#define INVERT_Y_DIR true  
#define INVERT_Z_DIR false
```

Ender5:

```
#define INVERT_X_DIR false  
#define INVERT_Y_DIR false  
#define INVERT_Z_DIR true
```

4. The size of the print bed Ender3:

```
#define X_BED_SIZE 220  
#define Y_BED_SIZE 220
```

Ender5:

```
#define X_BED_SIZE 235  
#define Y_BED_SIZE 235
```

and make sure to enable the noise threshold for endstop

```
#define ENDSTOP_NOISE_THRESHOLD 2
```

where-to-download--board-case

3D files of board

board V1.0 case stl for ender3 printer

board V2.5 case stl

anyone who changes or adds anything is appreciated.

How-to-install-pandapi-to-the-latest-octopi-image

1 , compile RT kernel and replace the old one in the image of octopi;

<https://www.instructables.com/64bit-RT-Kernel-Compilation-for-Raspberry-Pi-4B-/>

2, install ttyOtty driver:

<https://github.com/freemed/ttyOtty>

3, download the marlin compile&configure tools

```
cd /home/pi/phtml/
```

```
svn co https://github.com/markniu/doc_test.git/trunk/imges/html
```

4,download marlin source code

```
cd /home/pi/PandaPI/
```

```
svn co https://github.com/markniu/PandaPi.git/trunk/Marlin2.x
```

5,install pi_marlin

```
cd /home/pi/PandaPI/Marlin2.x/pandapi/
```

```
make clean;make -j4
```

```
sudo chmod 777 pi_marlin
```

```
cp pi_marlin /home/pi
```

6,add the autorun command in the start shell **/etc/rc.local**

```
node /home/pi/phtml/html/webserver.js &
```

```
/home/pi/phtml/html/monitor.sh &
```

7.configure auto mount USB stick

<https://raspberrypi.stackexchange.com/questions/66169/auto-mount-usb-stick-on-plug-in-without-uuid>

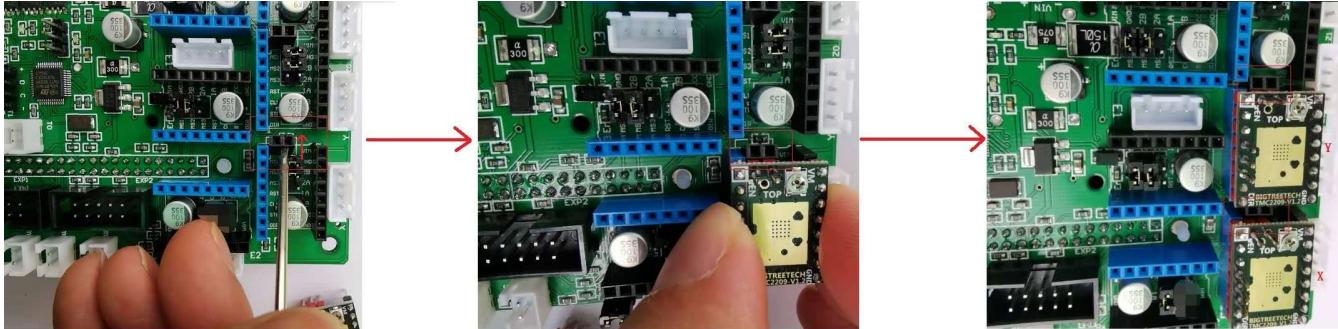
End, good luck! :)

FAQ

endstop not work?

1.check diag pin

If you use limit endstop instead of sensorless homing, the DIAG pin is not used, and not allowed to plug in, so you can bend it like the following or cut off the DIAG pin on the driver.



Disable Sensorless Home

If you use X/Y Endstop, please disable sensorless home

2.Get endstop status by Send gcode M119 through octoprint

```
Send: M119
Recv: Reporting endstop status
Recv: x_min: open
Recv: y_min: open
Recv: z_min: TRIGGERED
```

3.Endstops falsely trigger due to noise,

enable the noise threshold in Configuration.h

```
#define ENDSTOP_NOISE_THRESHOLD 2
```

TMC Error message:

```
Testing X connection... Error: All LOW
Recv: Testing Y connection... Error: All LOW
Recv: Testing Z connection... Error: All LOW
Recv: Testing E connection... Error: All LOW
```

The reason of this TMC Error message is that with write only access to TMC, and there is no RXD connection in the hardware. So that's normal.
[https://github.com/markniu/PandaPi/wiki/How-to-run-TMC2209-with-UART-mode-\(V2.0-2.5\)](https://github.com/markniu/PandaPi/wiki/How-to-run-TMC2209-with-UART-mode-(V2.0-2.5))

LCD12864 blank , but can be connected from octopi.

please check the LCD type in the configuration.h, if you use LCD12864 please disable the touch LCD by add "://" before the DGUS_LCD_UI_PANDAPI as following

```
//#define DGUS_LCD_UI_PANDAPI
```

