

3DTouch Auto Leveling Sensor

User Manual



1. Introduction

3DTouch is an auto leveling sensor for 3D Printers that can precisely measure the tilt of your print surface. It can greatly improve the printing precision of your 3D Printer.

3DTouch features simple, smart and precise. It could work with nearly any kind of bed materials, such as glasses, woods, metals and so on.

The main functions and controls of 3DTouch are the same as most auto bed leveling sensors, which consists of a RC servo and a micro switch, thus, 3DTouch can be used on almost every 3D printer control board.

By using progressively designed solenoid and hall sensor, 3DTouch can integrate high precision in such a simple structure. To make it more user-friendly and to bring you more enjoyable printing experience we add many smart functions such as self-test, false alarm, alarm release and test mode for M119.

Features:

1 Simple

3DTouch can be easily applied, since it has a small and simple structure. Gathering information & firmware setting will be an easy task, because 3DTouch works as usual auto bed leveling sensor.

2 Smart

Self-test: The push pin is operated three times to test when the power is on

Alarm: The LED light blinks if a problem found on a self-test or on an operation

3 High-precision

3DTouch's Standard Deviation in repeatability is around 0.005mm, at that precise.

If you choose 3DTouch, your 3D printer will be high-class masterpiece, giving you an enjoyable experience.

4. Innovative Solenoid: Ultra Power Saving

On idle state, while the push-pin is whether pulled out or retracted, there are not any electric current flowing on solenoid, and standby electric current in the whole device is below 15mA on average, whereas on working state, while the pin is moving in sudden about 100ms, under 300mA flows in the device.

Low power consumption even further drops joule heating, preventing from heat problem.

5 Technologies

3DTouch consists of Atmel ATtiny13A, solenoid, and a push pin.

6 wide Selection of Bed

3DTouch does not use either optical, nor proximity (inductive/capacitive) sensor.

3DTouch is controlled by Hall Effect, providing high precision. Thus the bed material can be selected freely.

7 Optimized structure: Larger Build Size

3DTouch is a small and technology-intensive one. Build size can be set larger than other existing auto bed leveling sensor.

3DTouch uses existing RC Servo motor signal intactly, so just plug 3DTouch on the same pins after removing servo motor.

Specifications:

Voltage: 5V

Current: 15mA

Max. Current: 300mA

Cable length: 150mm

Net Weight: 10g

Shipping weight: 25g

Wiring

3-pin: Brown (-, GND), Red (+5V), Orange (control signal)

2-pin: Black (-, GND), White (Z min)

Wiring:

Brown (-, GND)

Red (+5V)

Orange (control signal)

Black (-, GND)

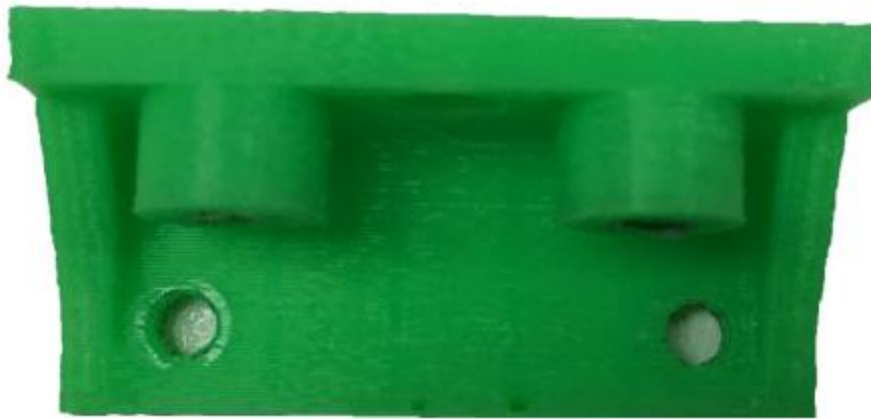
White (Z min)

**2. How to use it****2.1 mount the 3DTouch sensor**

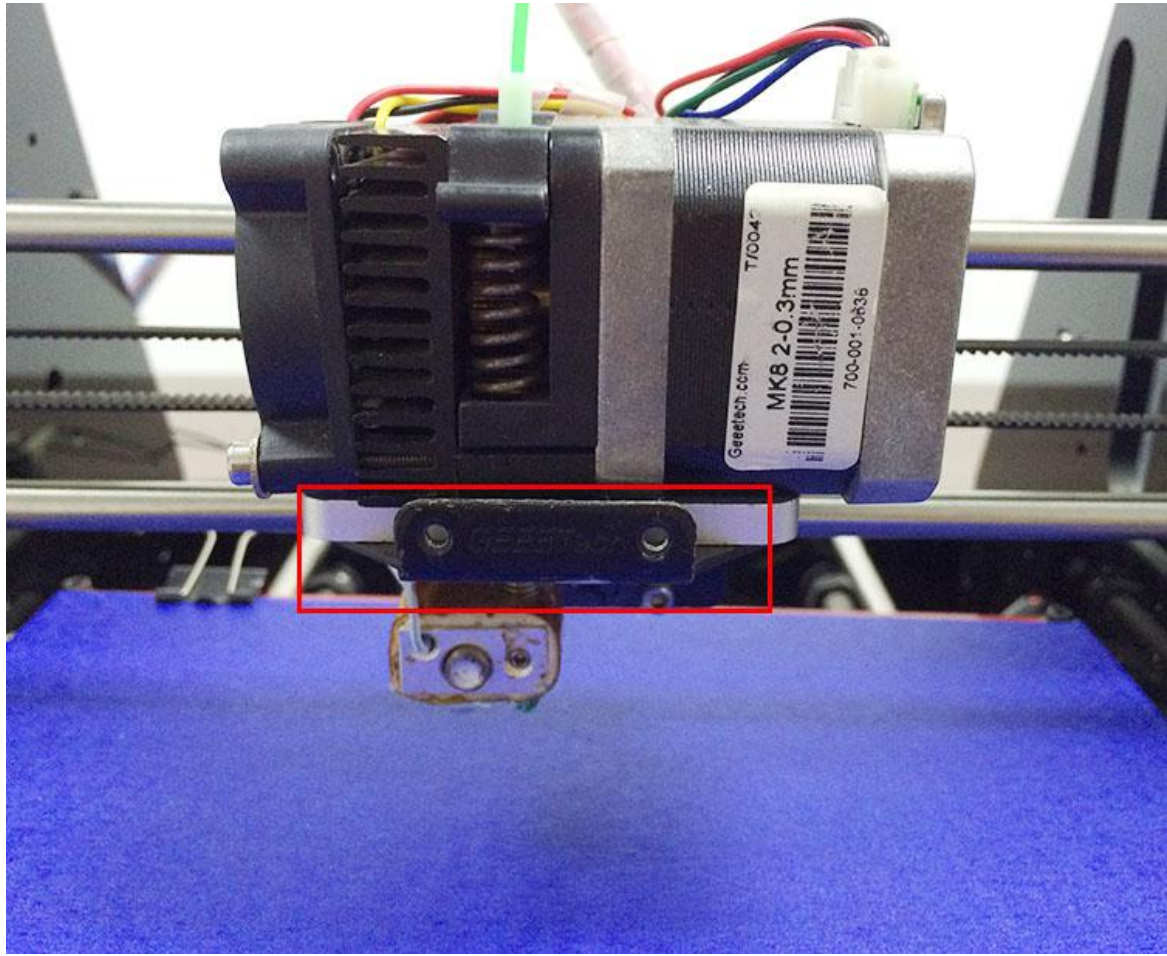
So far we have successfully tested our 3DTouch sensor on Geeetech Prusa I3 pro B, Pro C and Pro X.

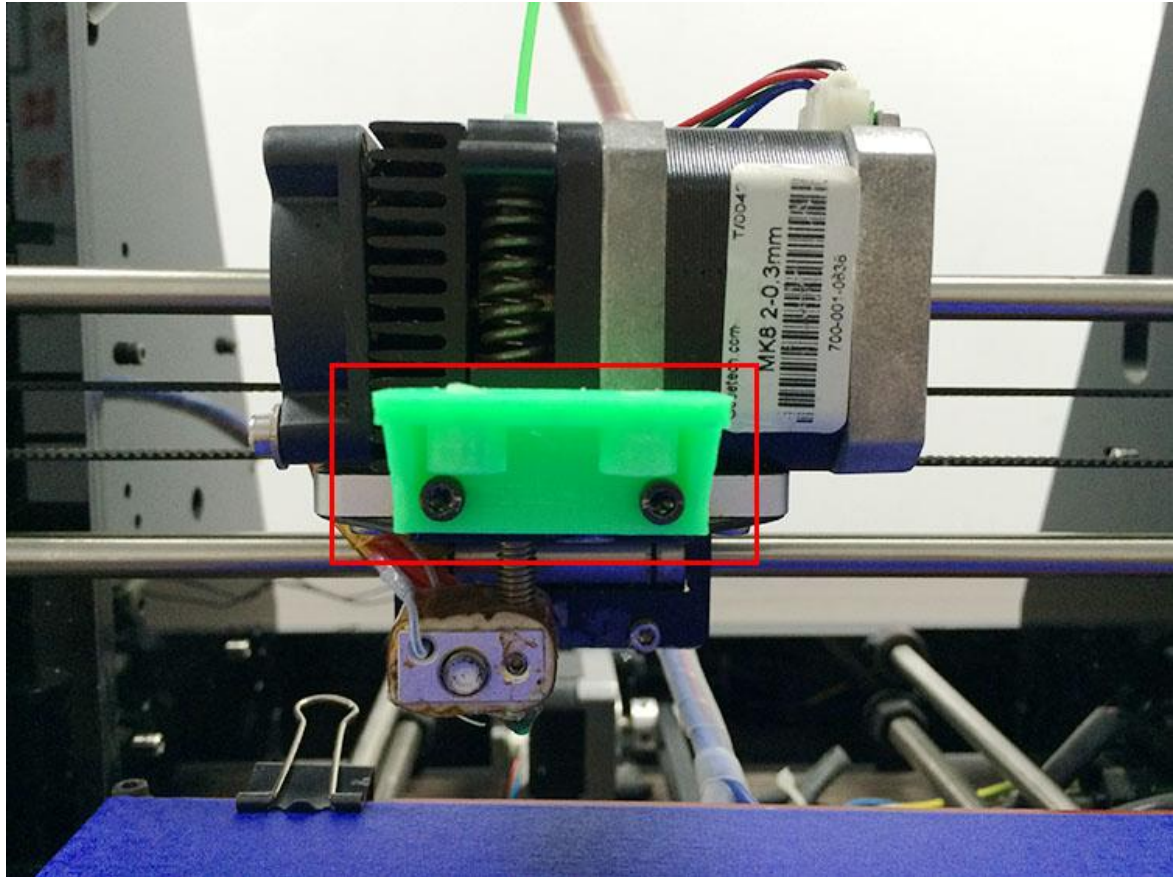
Here is a detailed instructions on how to use the 3DTouch sensor to your geeetech pro B. For pro C and pro X, the steps are the same.

You will need a suitable mount to attach the 3DTouch sensor to your printer. Here is a 3DTouch sensor mount:

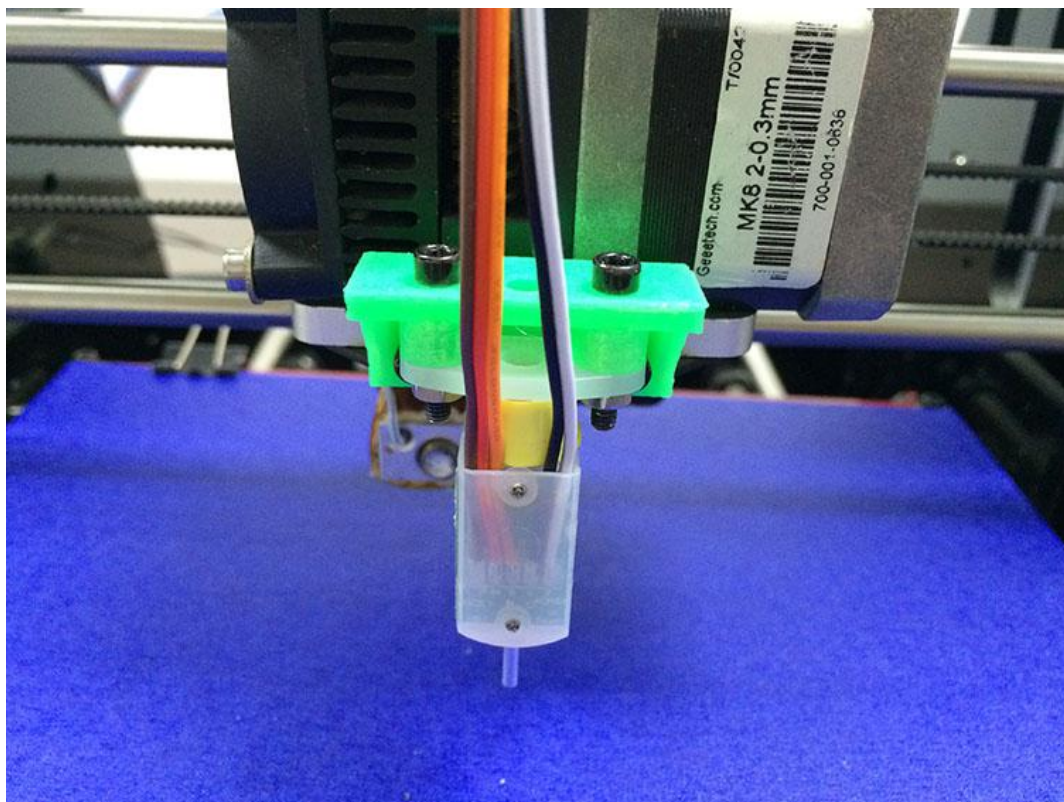
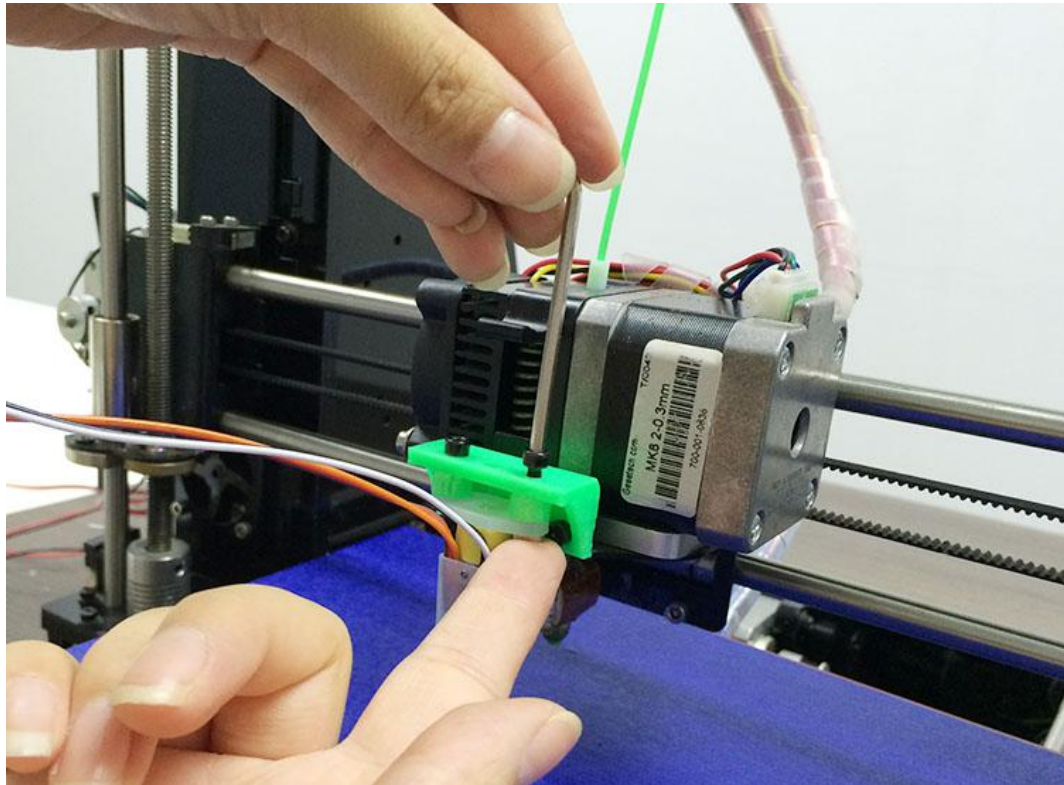


1. Download the .stl file [here](#) and print one.
2. Fix the mount on the Extruder holder with 2 M3*6mm screw.





3. Fix the 3D Touch sensor on the sensor mount with 2 M3*16mm screws and 2 M3 nuts.



2.2 Wiring

The 3DTouch Auto Leveling sensor has 5 wires, 3 for the first servo connection and 5v and 2 for the Z min end stop, negative and signal pins.

3DTouch can be operated in the following condition.

One I/O for control (PWM or Software PWM)

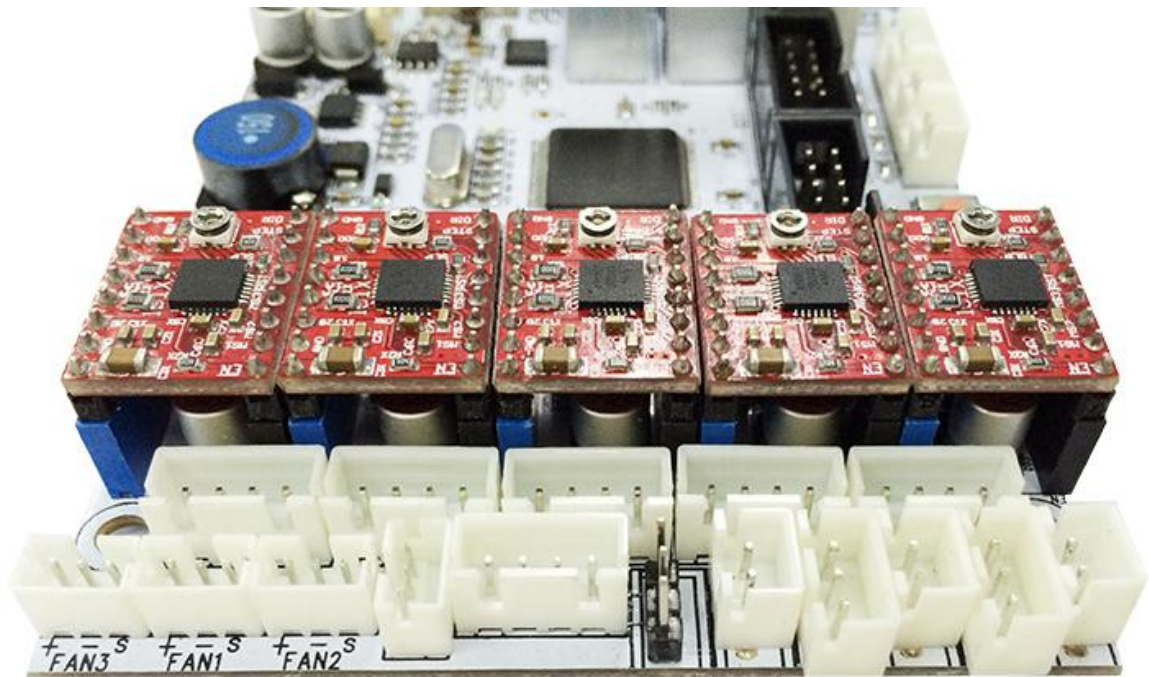
One I/O for Z min (Z Probe)

GND and +5V power

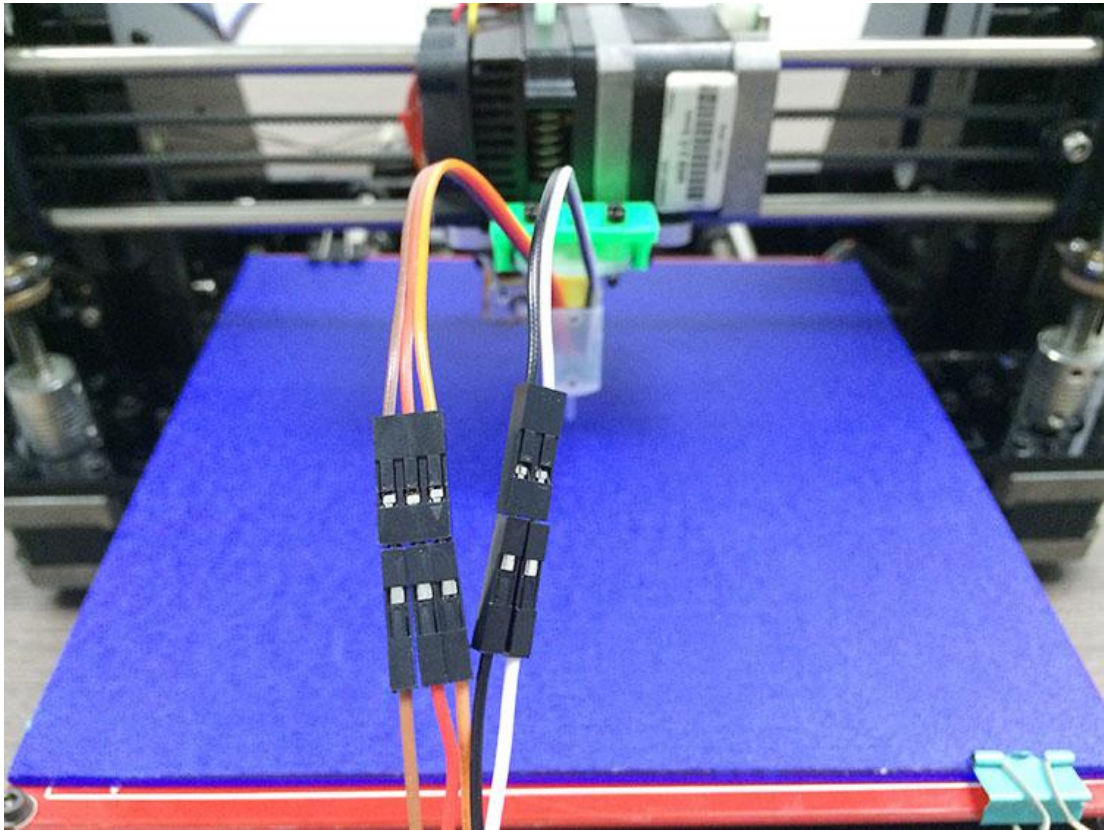
Let's take our geeetech GT2560 3D Printer control board as an example.

There are several ways to connect the 3DTouch Auto Leveling sensor to GT2560, here is the easiest way.

Step1. Remove the Z max connector from the board and replace it with a 3Pin Straight Pin. You need to use soldering iron here.



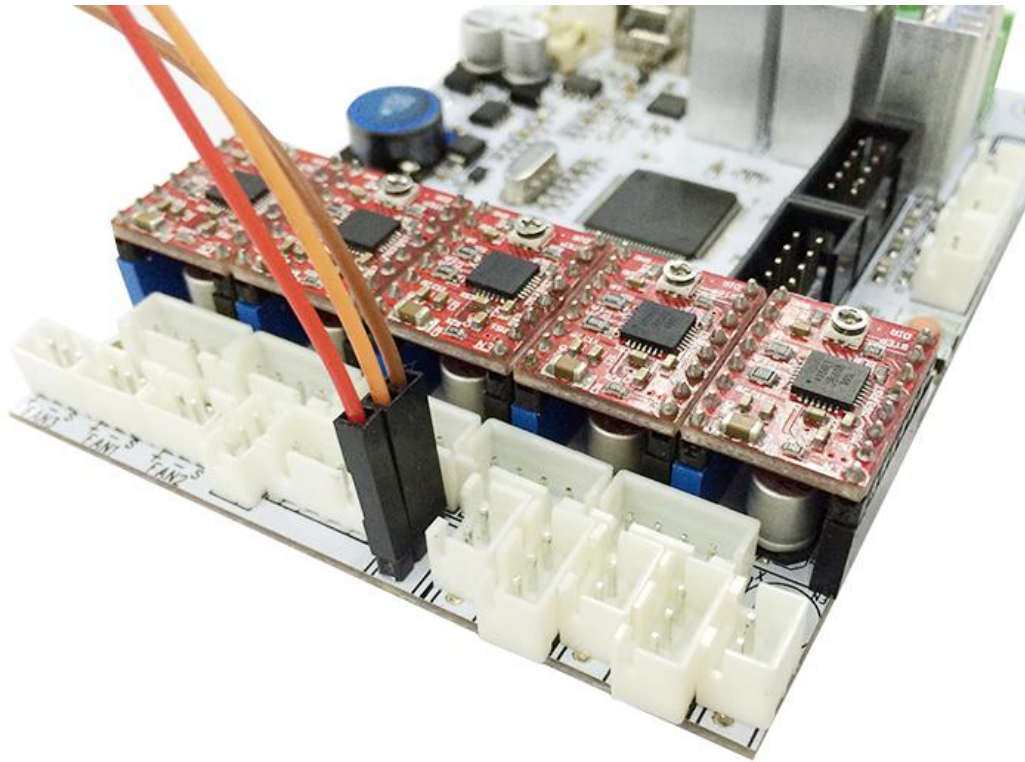
Step2. Use DuPont wire to extend the wires of 3DTouch. It doesn't matter if you cannot find the wires with the same color, but do not mix the wires up.



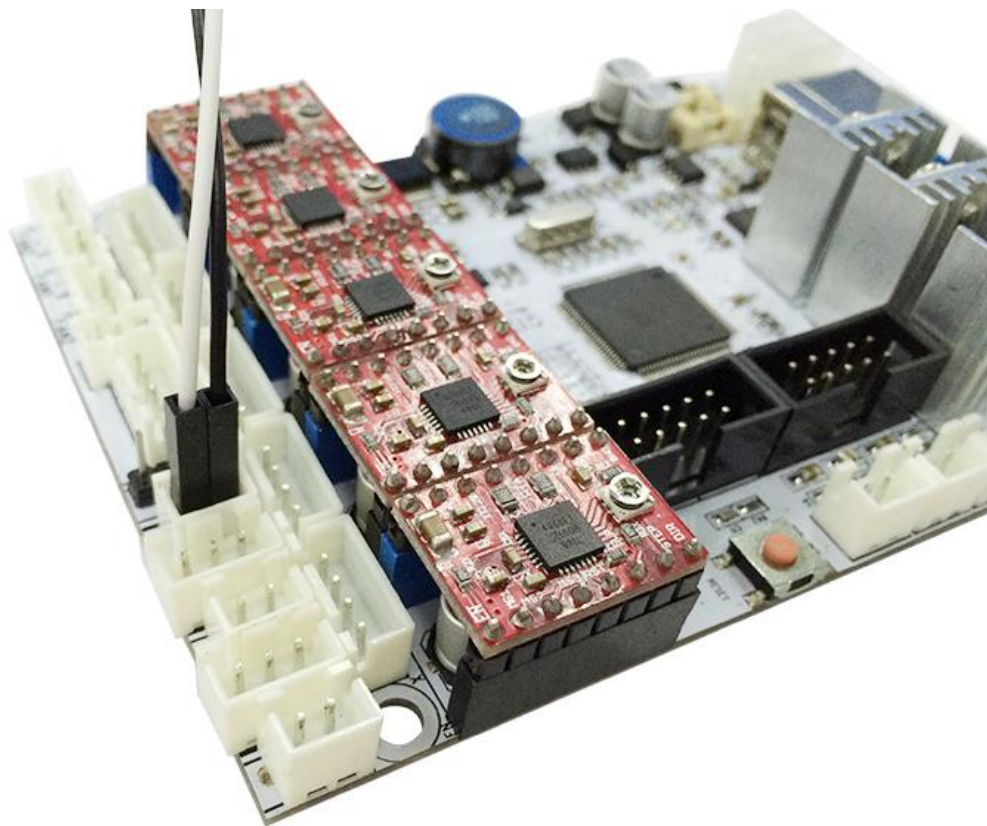
Step3. Connect the extended wire to the GT2560 control board.

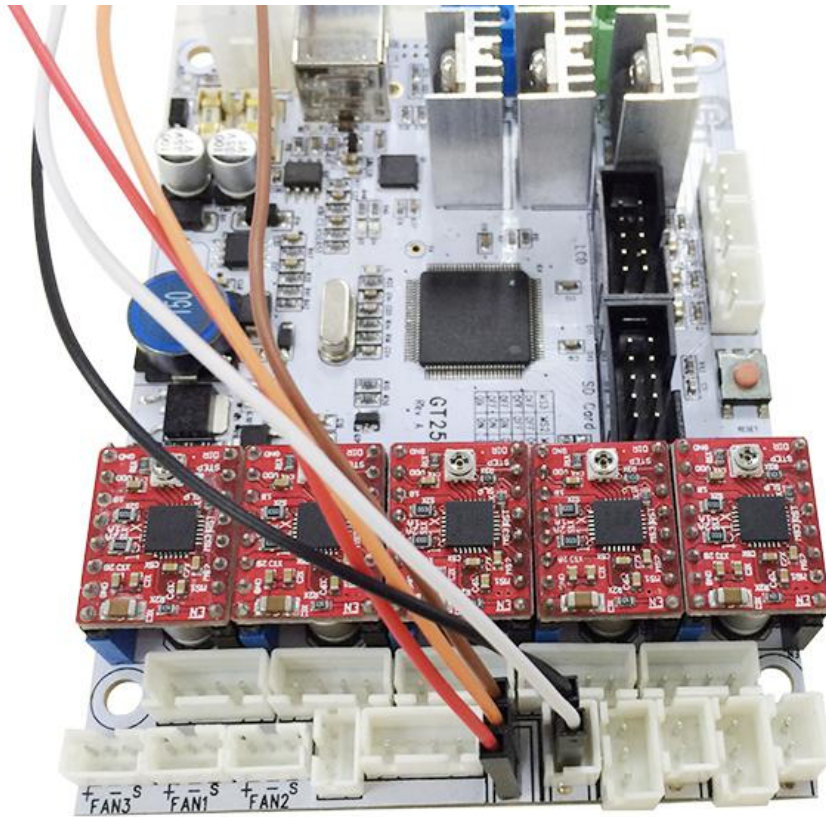
Connect the 3 pin wire to the Z max pin.

3-pin: Brown (-, GND) Red (+5V) Orange (control signal)



Connect the 2 pin wire to the Z min pin.





Note the wire order.

- When using 3D Touch Auto Leveling sensor, you do not need to connect the original Z min end stop wires.

That's all for the wiring of the 3D Touch Auto Leveling sensor and GT2560.

2.3. Firmware setting

Changes need to be made for the configuration file in the Marlin source code for 3D Touch. The required changes are similar to how you would setup a mechanical servo sensor.

Step1. Find the following code in *Configuration.h*:



```

/*****\
 * R/C SERVO support
 * Sponsored by TrinityLabs, Reworked by codexmas
 *****/

// Number of servos
//
// If you select a configuration below, this will receive a default value and does not need to be set manually
// set it manually if you have more servos than extruders and wish to manually control some
// leaving it undefined or defining as 0 will disable the servo subsystem
// If unsure, leave commented / disabled
//
// #define NUM_SERVOS 3 // Servo index starts with 0 for M280 command

// Servo Endstops
//
// This allows for servo actuated endstops, primary usage is for the Z Axis to eliminate calibration or bed height changes.
// Use M206 command to correct for switch height offset to actual nozzle height. Store that setting with M500.
//
// #define SERVO_ENDSTOPS {-1, -1, 0} // Servo index for X, Y, Z. Disable with -1
// #define SERVO_ENDSTOP_ANGLES {0,0, 0,0, 70,0} // X,Y,Z Axis Extend and Retract angles

```

Modify the code in the red box into:

```

// Number of servos
//
// If you select a configuration below, this will receive a default value and does not
need to be set manually
// set it manually if you have more servos than extruders and wish to manually control
some
// leaving it undefined or defining as 0 will disable the servo subsystem
// If unsure, leave commented / disabled
//
#define NUM_SERVOS 1 // Servo index starts with 0 for M280 command

// Servo Endstops
//
// This allows for servo actuated endstops, primary usage is for the Z Axis to eliminate
calibration or bed height changes.

```

// Use M206 command to correct for switch height offset to actual nozzle height.
Store that setting with M500.

//

#define SERVO_ENDSTOPS {-1, -1, 0} // Servo index for X, Y, Z. Disable with -1
#define SERVO_ENDSTOP_ANGLES {0,0, 0,0, 10,90} // X,Y,Z Axis Extend and
Retract angles

Step2. Find the codes regarding to Bed Auto Leveling in *Configuration.h*.



```
Marlin | Arduino 1.0.1
File Edit Sketch Tools Help

Marlin | BlinkM.cpp | BlinkM.h | Configuration.h | ConfigurationStore.cpp | ConfigurationStore.h | Configuration_adv.h | DOGMbitmaps.h

#define Z_MAX_LENGTH (Z_MAX_POS - Z_MIN_POS)
//===== Bed Auto Leveling =====

// #define ENABLE_AUTO_BED_LEVELING // Delete the comment to enable (remove // at the start of the line)
#define Z_PROBE_REPEATABILITY_TEST // If not commented out, Z-Probe Repeatability test will be included if Auto Bed Leveling is Enabled.

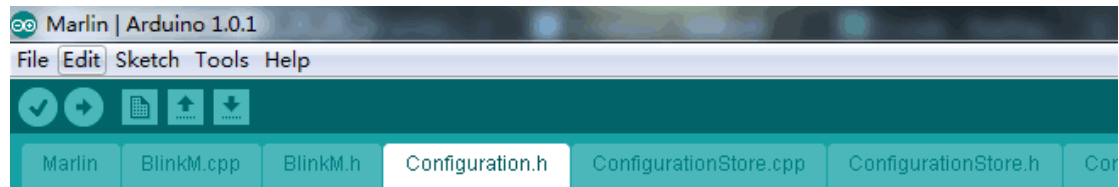
#ifdef ENABLE_AUTO_BED_LEVELING
|
// There are 2 different ways to pick the X and Y locations to probe:

// - "grid" mode
//   Probe every point in a rectangular grid
//   You must specify the rectangle, and the density of sample points
//   This mode is preferred because there are more measurements.
//   It used to be called ACCURATE_BED_LEVELING but "grid" is more descriptive

// - "3-point" mode
//   Probe 3 arbitrary points on the bed (that aren't colinear)
//   You must specify the X & Y coordinates of all 3 points

#define AUTO_BED_LEVELING_GRID
// with AUTO_BED_LEVELING_GRID, the bed is sampled in a
// AUTO_BED_LEVELING_GRID_POINTSxAUTO_BED_LEVELING_GRID_POINTS grid
// and least squares solution is calculated
// Note: this feature occupies 10'206 byte
#ifdef AUTO_BED_LEVELING_GRID

// set the rectangle in which to probe
#define LEFT_PROBE_BED_POSITION 15
#define RIGHT_PROBE_BED_POSITION 170
#define BACK_PROBE_BED_POSITION 180
#define FRONT_PROBE_BED_POSITION 20
```



```

// set the number of grid points per dimension
// I wouldn't see a reason to go above 3 (=9 probing points on the bed)
#define AUTO_BED_LEVELING_GRID_POINTS 2

#else // not AUTO_BED_LEVELING_GRID
// with no grid, just probe 3 arbitrary points. A simple cross-product
// is used to estimate the plane of the print bed

#define ABL_PROBE_PT_1_X 15
#define ABL_PROBE_PT_1_Y 180
#define ABL_PROBE_PT_2_X 15
#define ABL_PROBE_PT_2_Y 20
#define ABL_PROBE_PT_3_X 170
#define ABL_PROBE_PT_3_Y 20

#endif // AUTO_BED_LEVELING_GRID

// these are the offsets to the probe relative to the extruder tip (Hotend - Probe)
// X and Y offsets must be integers
#define X_PROBE_OFFSET_FROM_EXTRUDER -25
#define Y_PROBE_OFFSET_FROM_EXTRUDER -29
#define Z_PROBE_OFFSET_FROM_EXTRUDER -12.35

#define Z_RAISE_BEFORE_HOMING 4 // (in mm) Raise Z before homing (G28) for Probe Clearance.
// Be sure you have this distance over your Z_MAX_POS in case

//=====Bed Auto
Leveling=====

```

#define ENABLE_AUTO_BED_LEVELING // Delete the comment to enable
(remove // at the start of the line)

#define Z_PROBE_REPEATABILITY_TEST // If not commented out, Z-Probe Repeatability test will be included if Auto Bed Leveling is Enabled.

#ifndef ENABLE_AUTO_BED_LEVELING

...

#define AUTO_BED_LEVELING_GRID

...

#ifndef AUTO_BED_LEVELING_GRID

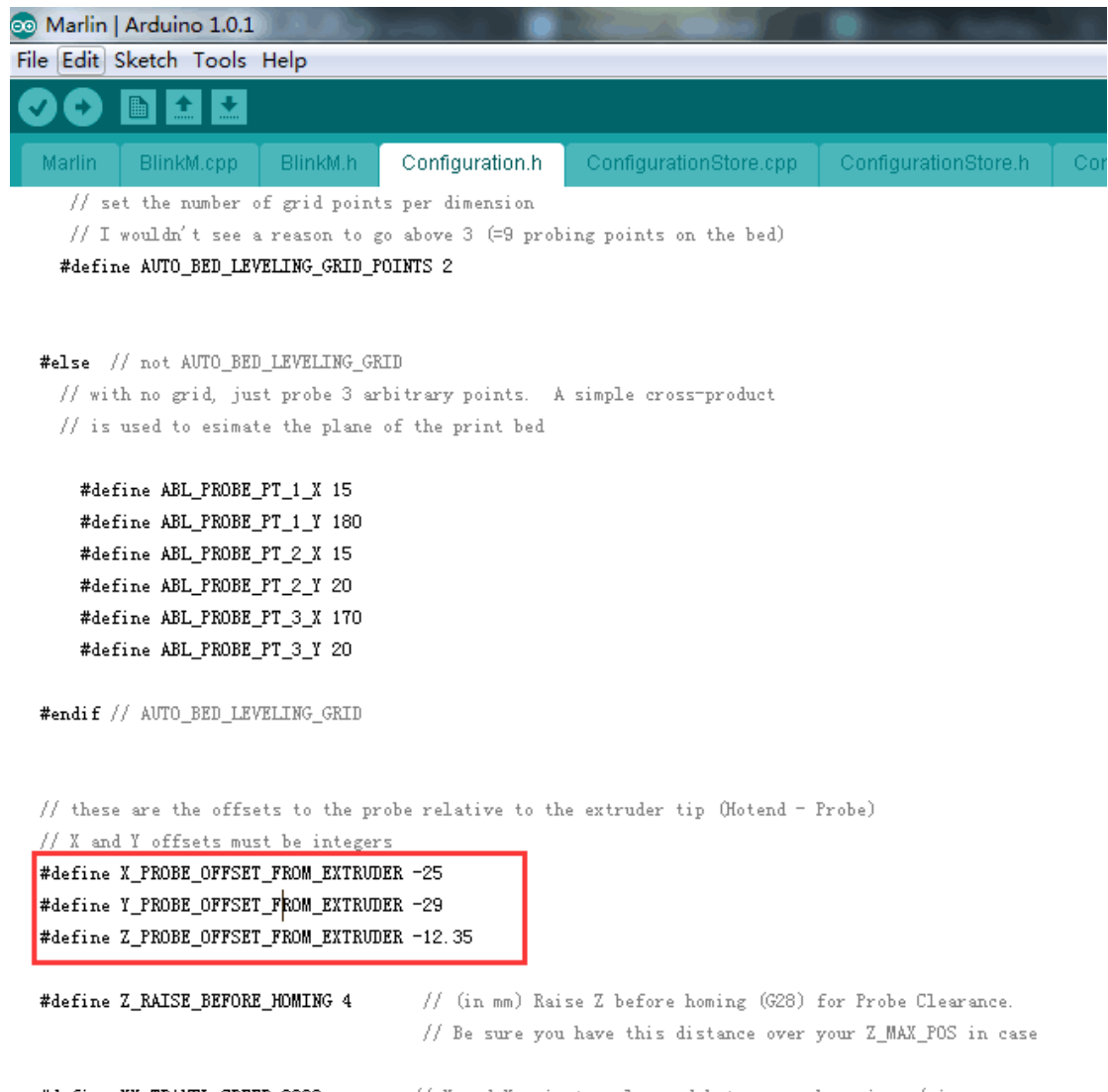
// set the rectangle in which to probe

#define LEFT_PROBE_BED_POSITION 30

#define RIGHT_PROBE_BED_POSITION 200

```
#define BACK_PROBE_BED_POSITION 147
#define FRONT_PROBE_BED_POSITION 20
```

Step3: scroll down to find the codes to Define the probe offset



```
Marlin | Arduino 1.0.1
File Edit Sketch Tools Help

Marlin BlinkM.cpp BlinkM.h Configuration.h ConfigurationStore.cpp ConfigurationStore.h Cor

// set the number of grid points per dimension
// I wouldn't see a reason to go above 3 (=9 probing points on the bed)
#define AUTO_BED_LEVELING_GRID_POINTS 2

#else // not AUTO_BED_LEVELING_GRID
// with no grid, just probe 3 arbitrary points. A simple cross-product
// is used to estimate the plane of the print bed

#define ABL_PROBE_PT_1_X 15
#define ABL_PROBE_PT_1_Y 180
#define ABL_PROBE_PT_2_X 15
#define ABL_PROBE_PT_2_Y 20
#define ABL_PROBE_PT_3_X 170
#define ABL_PROBE_PT_3_Y 20

#endif // AUTO_BED_LEVELING_GRID

// these are the offsets to the probe relative to the extruder tip (Hotend - Probe)
// X and Y offsets must be integers
#define X_PROBE_OFFSET_FROM_EXTRUDER -25
#define Y_PROBE_OFFSET_FROM_EXTRUDER -29
#define Z_PROBE_OFFSET_FROM_EXTRUDER -12.35



























#define Z_RAISE_BEFORE_HOMING 4 // (in mm) Raise Z before homing (G28) for Probe Clearance.
// Be sure you have this distance over your Z_MAX_POS in case

// ...
```

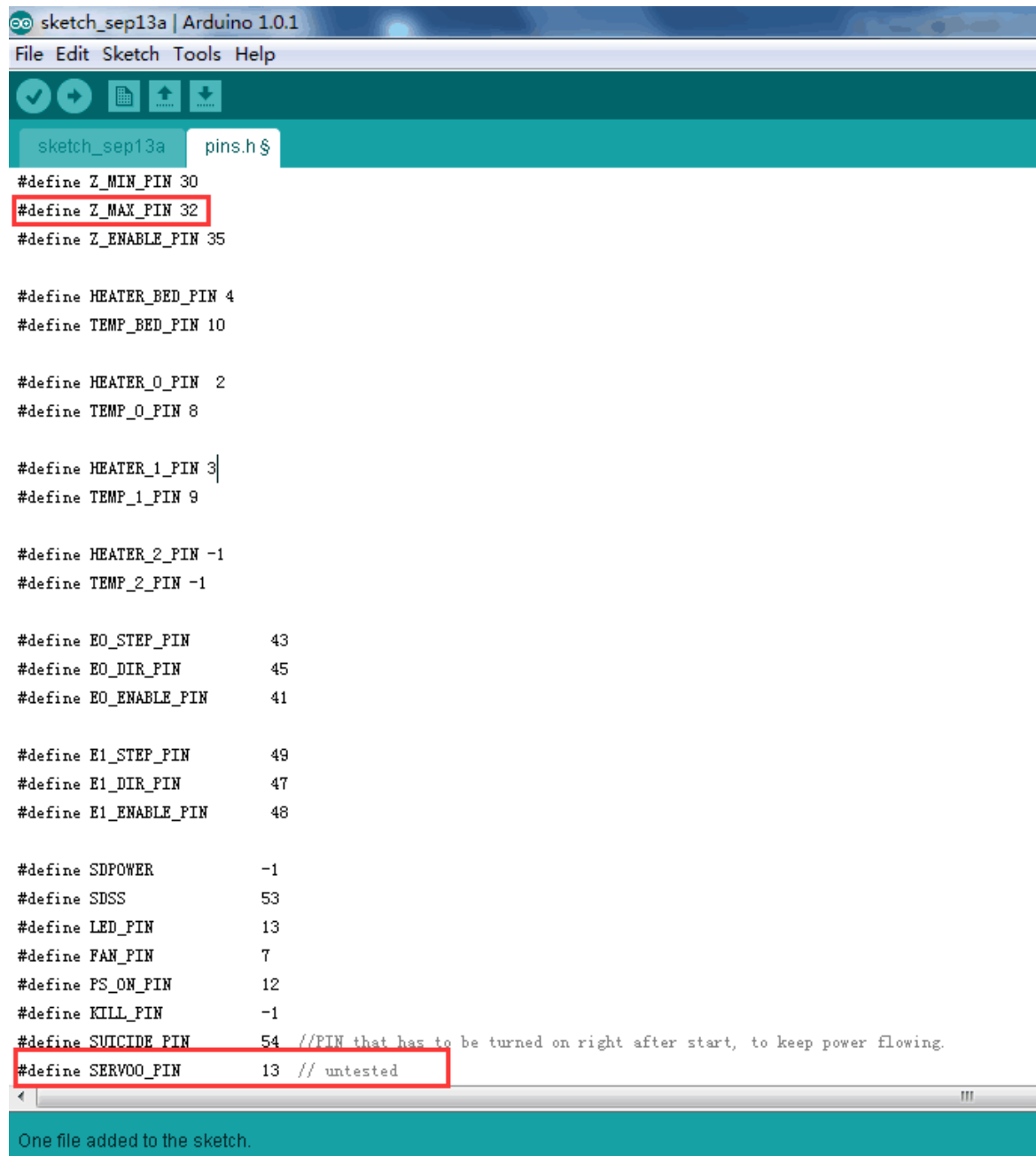
```
#define AUTO_BED_LEVELING_GRID_POINTS 2
#else // not AUTO_BED_LEVELING_GRID
...
#define X_PROBE_OFFSET_FROM_EXTRUDER 6
#define Y_PROBE_OFFSET_FROM_EXTRUDER -43
#define Z_PROBE_OFFSET_FROM_EXTRUDER -1.4
```

Step4. Find the following code in pins.h

If you do not find the *pins.h* tab on Arduino IDE, please open it separately, after the modification, please save it.

	language_fi.h	2016/9/13 15:11	H 文件	8 KB
	language_fr.h	2016/9/13 15:11	H 文件	8 KB
	language_it.h	2016/9/13 15:11	H 文件	8 KB
	language_nl.h	2016/9/13 15:11	H 文件	8 KB
	language_pl.h	2016/9/13 15:11	H 文件	8 KB
	language_pt.h	2016/9/13 15:11	H 文件	8 KB
	language_ru.h	2016/9/13 15:11	H 文件	9 KB
	LiquidCrystalRus.cpp	2016/9/13 15:11	CPP 文件	11 KB
	LiquidCrystalRus.h	2016/9/13 15:11	H 文件	4 KB
	Marlin.h	2016/9/13 15:11	H 文件	9 KB
	Marlin.pde	2016/9/13 15:11	PDE 文件	2 KB
	Marlin_main.cpp	2016/9/13 15:11	CPP 文件	153 KB
	Marlin1.ino	2016/9/13 15:11	Arduino Source ...	2 KB
	MarlinSerial.cpp	2016/9/13 15:11	CPP 文件	8 KB
	MarlinSerial.h	2016/9/13 15:11	H 文件	6 KB
	motion_control.cpp	2016/9/13 15:11	CPP 文件	7 KB
	motion_control.h	2016/9/13 15:11	H 文件	2 KB
	pins.h	2016/9/13 15:11	H 文件	80 KB
	planner.cpp	2016/9/13 15:11	CPP 文件	42 KB
	planner.h	2016/9/13 15:11	H 文件	7 KB
	qr_solve.cpp	2016/9/13 15:11	CPP 文件	41 KB
	qr_solve.h	2016/9/13 15:11	H 文件	2 KB
	Sd2Card.cpp	2016/9/13 15:11	CPP 文件	23 KB
	Sd2Card.h	2016/9/13 15:11	H 文件	9 KB
	Sd2PinMap.h	2016/9/13 15:11	H 文件	13 KB
	SdBaseFile.cpp	2016/9/13 15:11	CPP 文件	55 KB

Find the code as shown in the red box:



```
sketch_sep13a | Arduino 1.0.1
File Edit Sketch Tools Help

sketch_sep13a pins.h$
#define Z_MIN_PIN 30
#define Z_MAX_PIN 32
#define Z_ENABLE_PIN 35

#define HEATER_BED_PIN 4
#define TEMP_BED_PIN 10

#define HEATER_0_PIN 2
#define TEMP_0_PIN 8

#define HEATER_1_PIN 3
#define TEMP_1_PIN 9

#define HEATER_2_PIN -1
#define TEMP_2_PIN -1

#define EO_STEP_PIN      43
#define EO_DIR_PIN       45
#define EO_ENABLE_PIN    41

#define E1_STEP_PIN      49
#define E1_DIR_PIN       47
#define E1_ENABLE_PIN    48

#define SDPOWER          -1
#define SDSS             53
#define LED_PIN          13
#define FAN_PIN          7
#define PS_ON_PIN        12
#define KILL_PIN         -1
#define SUICIDE_PIN       54 //PIN that has to be turned on right after start, to keep power flowing.
#define SERVOO_PIN        13 // untested

One file added to the sketch.
```

```
/*
*****
* Ultimaker pin assignment
*****
*/
```

```
#if MB(ULTIMAKER)
#define KNOWN_BOARD

...
#define Z_MAX_PIN -1//32
#define Z_ENABLE_PIN 35
```

```
...  
#define SUICIDE_PIN      54  //PIN that has to be turned on right after start, to  
keep power flowing.  
#define SERVO0_PIN      32//13  // untested
```

Now, we have finished the firmware; please upload the modified firmware to your control board.

3. Testing

When the 3DTouch is first powered up it does a self test – Starting with the pin up it then goes down/up 3 times and ends up the LED on solid. Continuous flashing means that there is an obstruction or fault.

The 3DTouch acts on the following g.code that can be used manually to diagnose faults etc but you don't need to normally worry about them.

M280 P0 S10 ; pushes the pin down

M280 P0 S90 ; pulls the pin up

M280 P0 S120 ; Self test – keeps going until you do pin up/down or release alarm

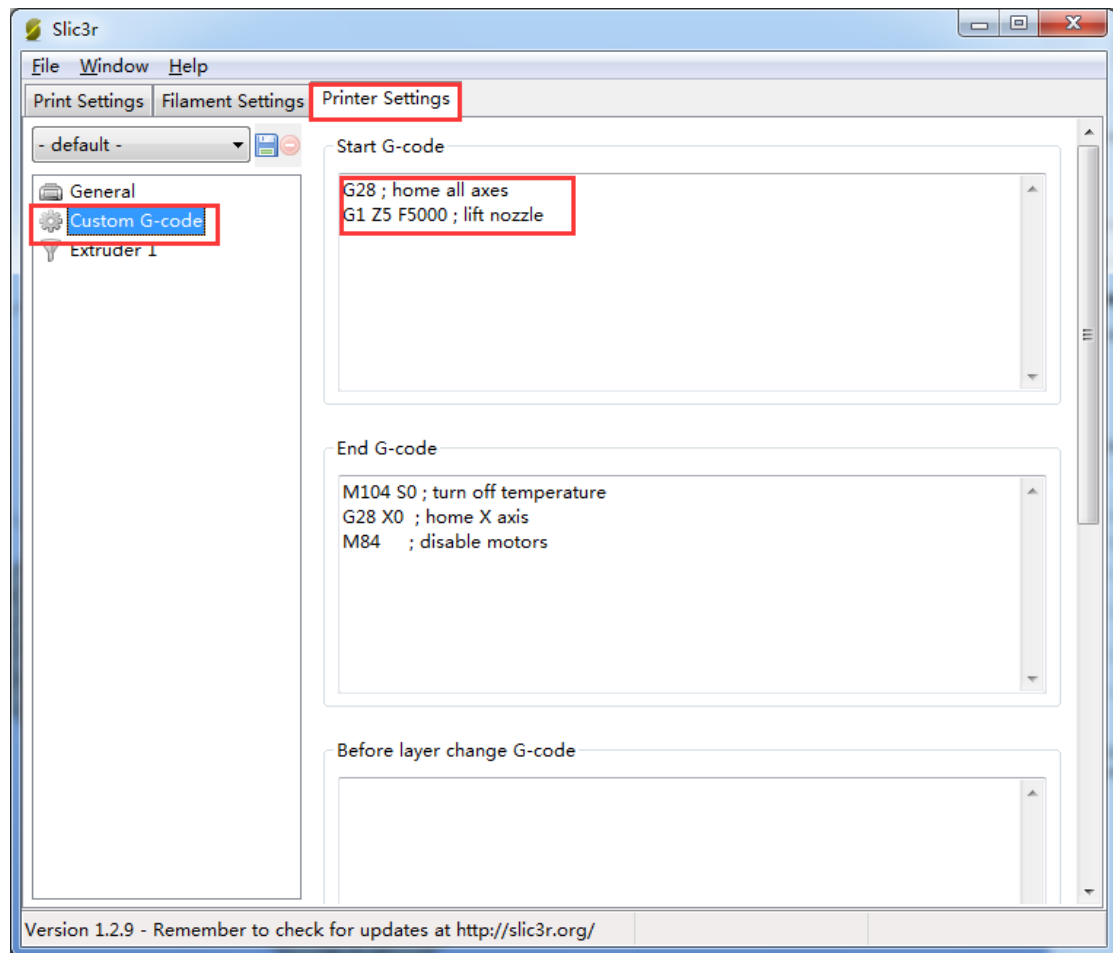
M280 P0 S160 ; Release alarm

Alarm – The 3DTouch can sense when something is wrong and then goes into alarm mode which is continuous flashing. Alarm can be triggered like an obstruction that stops the pin going up and down freely, it could be dirt etc.

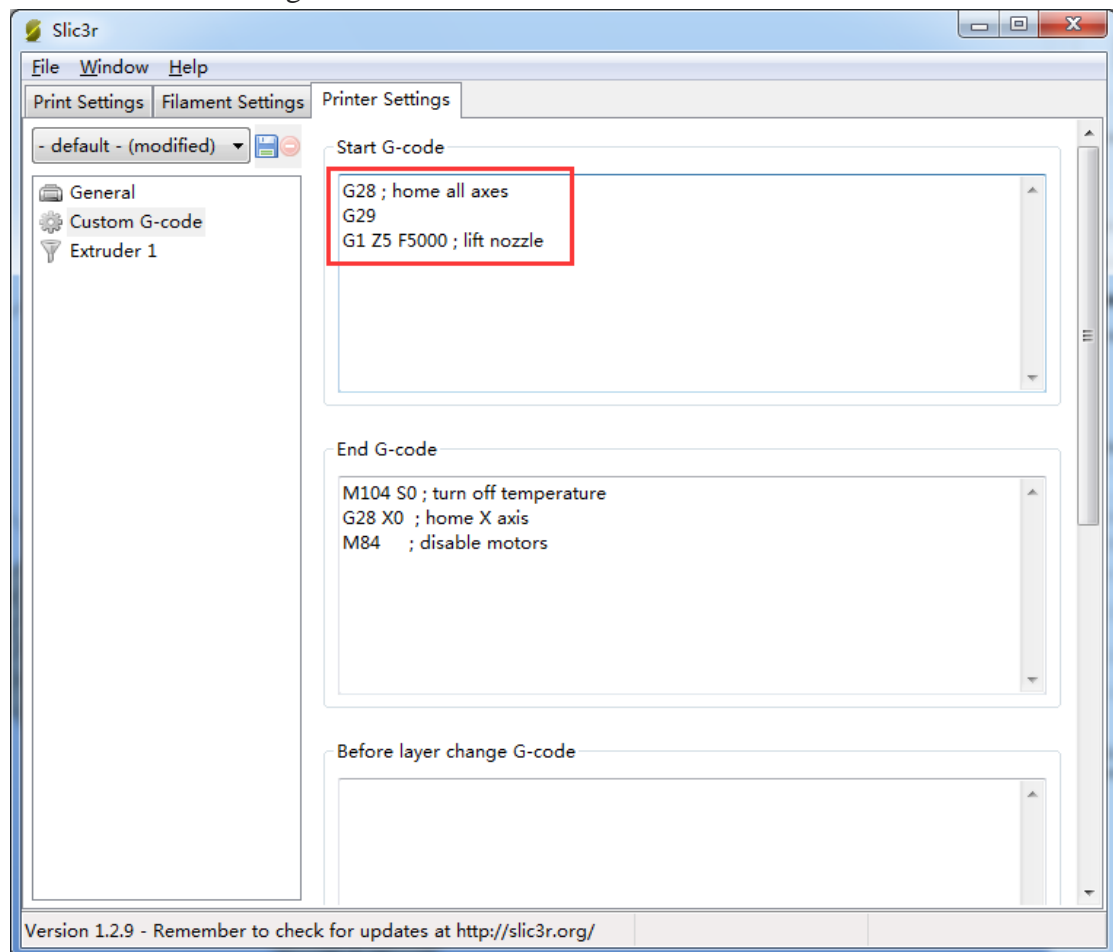
4. Printer setting

Providing the firmware is correctly configured, the sensor responds to the same codes as any other sensor e.g. inductive, capacitive or IR. The Start Code in your slicer should contain the sequence G28 followed by G29 to do the auto bed leveling.

Open Slicer>printer setting



Add G29 command right after G28



*Don't put another G28 after the G29 as it will just remove the G29 results.

The G29 command should be added every time.

[Here](#) is a video of using the 3DTouch Auto Leveling Sensor on Geeetech Prusa I3 pro B 3d printer.
