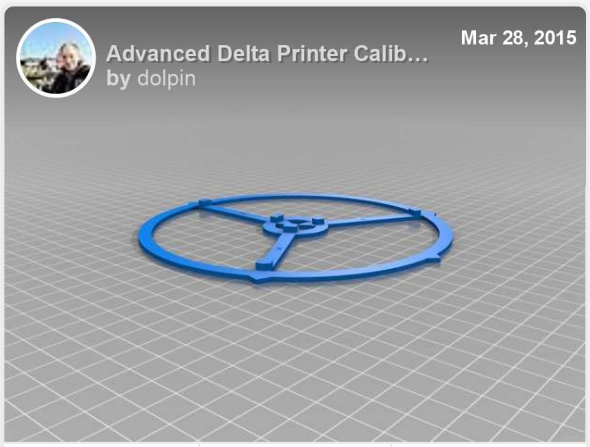
Thingiverse

Hey! This thing is still a Work in Progress. Files, instructions, and other stuff might change!



269 | 434 | 40

2 | Share

Download This Thing!

THING DETAILS

Thing Info

21706 Views

3990 Downloads

Found in 3D Printing Tests

Report Thing

Summary

This calibration object will help you to calibrate angle and size errors on your delta 3d printer

These errors are typically caused by tower po-

sition errors and different diagonal rod lengths.

This calibration object will show if your printer suffers from sizing and angle errors and the instructions will help to solve these errors.

My Achatz easyDelta printer had a 1 degree error in angles and over 1mm error in sizing in certain directions. This is caused by the towers not being positioned at a perfect 120 degree angle around the center, and small individual size errors between the rods.

First errors must be fixed on the printer itself, but on some printers this might not be possible.

I used a jig to glue the joints to the diagonal rods, but small errors (1/10 mm) result in bigger errors in the print.

The basic delta printer calibration steps are:

- calibrate the print bed on each tower position by adjusting the z-height for each carriage
- correct for bowl/dome shape by adjusting the DELTA_RADIUS parameter
- tune the print size by adjusting the DIAGONAL_ROD parameter

After the basic calibration you might experience that the actual print is not equal sized in all directions and the angles in all directions are not as they should be. Resulting in for instance trapezoid objects instead of square-angle objects.

The next calibration step is to detect these errors and calibrate your printer.

These errors are typically caused by tower misplacement and differences in diagonal rod length.

After calibrating the tower angles and individual diagonal rod lengths using the marlin firmware, I managed to achieve 0 degree angle error and a maximum of 0.1 mm size error in any direction in the print !

The calibration object is printed after the basic

calibration. Print out the calibration sheet and use this sheet to measure the angles between the X-Y-Z towers. Use a slide caliper to measure the print size in the X-Y-Z tower direction and calibrate your printer in the Marlin firmware.

Check the instructions for the calibration steps and how to correct this.

UPDATE
20-JUL-15: changed calibration object and added a calibration sheet
11-MAR-15: corrected an error regarding the angles in the drawing with instructions
06-SEP-15: corrected the diagonal rod calculation in the instructions

Instructions

Print the calibration object in PLA with 0.2mm layer height and 30% infill using a slow print speed (max 30mm/s).

Do not use ABS as ABS shrinks (0.7%) when cooling.

The 3 towers on a common delta printer are 120 degrees apart at a certain distance from the center (0,0) .

The calibration object will show tower misplacement errors and differences between the diagonal rods to the x, y, z carriage.

Prerequisites

- Make sure your towers are square and rigid (i used a water level to set the printbed level and measure/correct each tower using the water level)
- Have adjustable endstops on each carriage and a accurate endstop switch (optical or switch without lever) mounted firmly to your frame/rod

carriage with adjustable z-height: **http://www.thingiverse.com/thing:738217**

rigid end stop holder: **http://www.thingiverse.com/thing:642363**

rigid easyDelta 3d printer towers:
http://www.thingiverse.com/thing:746202

Instructions:

1. make sure your towers are square and rigid
2. having adjustable endstops on each tower makes calibration easier
3. calibrate your printer to the print bed by using the adjustable endstops and the DELTA_RADIUS parameter (marlin?: configuration.h) to correct for bowl/dome shape, see note below
4. print the calibration object
5. measure the angles between the x,y,z tower markers using the calibration sheet (120 degree). The sum of the angles must be 360 degrees
6. measure the print sizes in each tower direction using a sliding caliper (60mm)
7. correct the angles and sizes in the marlin software, see below
8. return to step 3 to check the calibration and recalibrate if required

The changes for the tower angles and individual rod length are done in the marlin_main.cpp (Marlin v1.0.2, see below for Marlin v1.1.0-RC2)

Angle correction

look for the definitions for COS_60 and SIN_60 and add definitions for 58, 59, 61 and 62 degree

```
#   define SIN_58 0.8480480961564259
#   define COS_58 0.5299192642332049
#   define SIN_59 0.8571673007021122
#   define COS_59 0.5150380749100542
#   define SIN_60 0.8660254037844386
#   define COS_60 0.5
#   define SIN_61 0.8746197071393958
#   define COS_61 0.4848096202463370
#   define SIN_62 0.8829475928589269
#   define COS_62 0.4694715627858907
```

On my printer the angle between x-y was 120, y-z was 121 and z-x was 119 degree.

Use the Z tower as a base and correct the tower position for the X and Y tower (see pictures for explanation):

- X-Z angle is 119: therefore change the tower 1 (x) angle in the firmware to 59 degree
- Y-Z angle is 121: therefore change the tower 2 (y) angle in the firmware to 61 degree

The towers angles is used in 2 different places in the marlin_main.cpp and you best can change at both places

```
delta_tower1_x= -SIN_59*delta_radius;
delta_tower1_y= -COS_59*delta_radius;
delta_tower2_x=  SIN_61*delta_radius;
delta_tower2_y= -COS_61*delta_radius;
delta_tower3_x= 0.0;
delta_tower3_y= delta_radius;
```

Size correction

Measure the x,y,z leg on the printed object and note down the measurements. Use the measured size most near to the required size to correct the general print size using the DELTA_DIAGONAL_ROD value in the configuration.h. See the note below on how to correct this

Now recalculate the print size for the other measurements by subtracting the general error from the measured sizes and use these values to correct the diagonal rod for each individual rod.

Look for the next line in marlin_main.cpp:

```
 float delta_diagonal_rod_2  = sq(delta_dia
gonal_rod);
```

and add a squared diagonal rod definition for each rod including the size correction

```
 float delta_diagonal_rod_2_x= sq(delta_dia
gonal_rod*(60.25/60));
 float delta_diagonal_rod_2_y= sq(delta_dia
```

```
gonal_rod*(60.31/60));
 float delta_diagonal_rod_2_z= sq(delta_dia
gonal_rod);
```

Look for the function: calculate_delta in the marlin_main.cpp file and replace the calculation with the code below:

```
delta[X_AXIS] = sqrt((delta_diagonal_rod_2_
x)
                     - sq(delta_tower1_x-cart
esian[X_AXIS])
                     - sq(delta_tower1_y-cart
esian[Y_AXIS])
                     ) + cartesian[Z_AXIS];
delta[Y_AXIS] = sqrt((delta_diagonal_rod_2_
y)
                     - sq(delta_tower2_x-cart
esian[X_AXIS])
                     - sq(delta_tower2_y-cart
esian[Y_AXIS])
                     ) + cartesian[Z_AXIS];
delta[Z_AXIS] = sqrt((delta_diagonal_rod_2_
z)
                     - sq(delta_tower3_x-cart
esian[X_AXIS])
                     - sq(delta_tower3_y-cart
esian[Y_AXIS])
                     ) + cartesian[Z_AXIS];'
```

I will use my measurements as an example:

- x size: 60.67mm
- y size: 60.73mm
- z size: 60.42mm

The z size is most accurate and therefore I used this measurement to correct the general DELTA_DIAGONAL_ROD (configuration.h)

```
sqrt((308.7^2) * (60.42/60)) = 309.78
```

(thanks to **http://www.thingiverse.com/clefranc** for helping me with the correct diagonal rod calculation)

Then I recalculated the print size (z size should have 0 error now)

- x size: 60.67 - 0.42 = 60.25 mm
- y size: 60.73 - 0.42 = 60.31 mm
- z size: 60.42 - 0.42 = 60.00 mm

And I used the recalculated print size to correct each individual rod length

```
 float delta_diagonal_rod_2_x= sq(delta_dia
gonal_rod*(60.25/60));
 float delta_diagonal_rod_2_y= sq(delta_dia
gonal_rod*(60.31/60));
 float delta_diagonal_rod_2_z= sq(delta_dia
gonal_rod);
```

note:
DELTA_RADIUS increase for dome shape and decrease for bowl shape. The center (0,0) height will remain the same and the height at the outer positions will increase/decrease

DIAGONAL_ROD increase if actual print size too big and decrease if too small. Use the following formula to correct the diagonal_rod value:

```
new_diagonal_rod = sqrt((diagonal_rod^2) *
(measured_print_size/required_print_size))
```

Marlin v1.0.2
The above instructions are all based on the Marlin firmware. I used v1.0.2 when I made this instruction.

Marlin v1.1.0-RC2 (thanks to http://www.thingiverse.com/GrAndAG)
In recent version (1.1.0-RC2) you can just define in Configuration.h:

```
#define DELTA_DIAGONAL_ROD_TRIM_TOWER_1 (DE
LTA_DIAGONAL_ROD*60.3/60 - DELTA_DIAGONAL_R
OD)
#define DELTA_DIAGONAL_ROD_TRIM_TOWER_2 (DE
LTA_DIAGONAL_ROD*60.05/60 - DELTA_DIAGONAL_
ROD)
#define DELTA_DIAGONAL_ROD_TRIM_TOWER_3 (DE
LTA_DIAGONAL_ROD*60/60 - DELTA_DIAGONAL_ROD
) // Means zero :)
```

And no other firmware modification is needed anymore.

Also you can use "M665 A### B### C###" to modify these values on the fly.

The tower angles can be indirectly adjusted via DELTA_RADIUS_TRIM_TOWER_1 or 2, 3 defines.

More from 3D Printing Tests

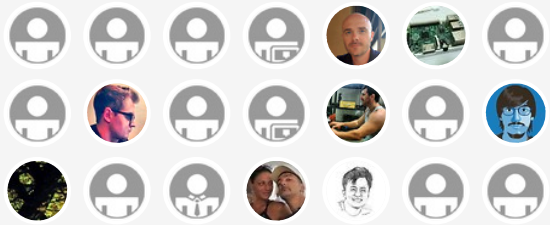
view more



Makes


Liked By

View All



Tags

calibration deltabot delta_printer easydelta printer_calibration

License