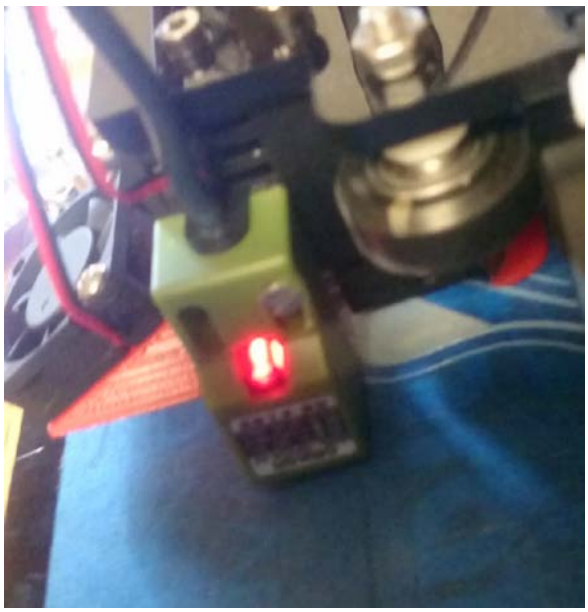
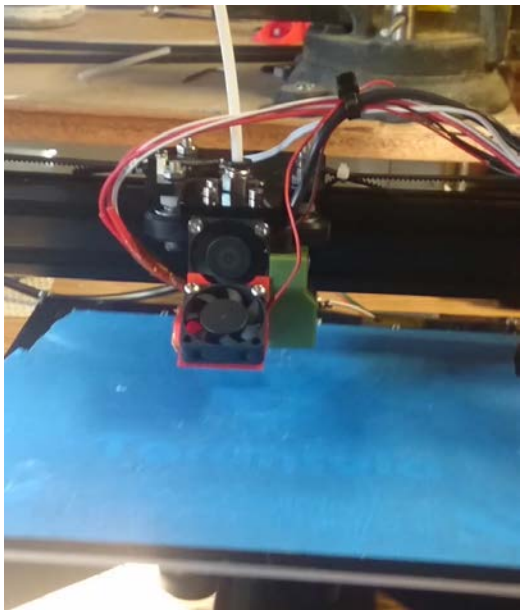
Tevo Tarantula Auto Level Instructions

By ArcadEd

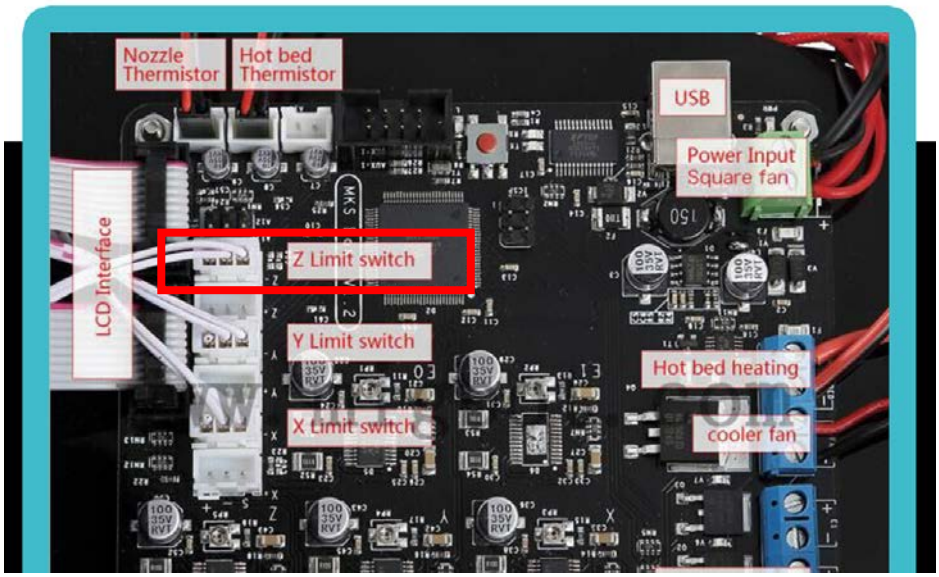
Mounting the Sensor:

First off, the kit does not seem to come with a screw for mounting the sensor. Maybe it does now, but at the time I got my kits, it did not. So you can either tap some threads into the side of the heatsink, or use a sheet metal screw like I did to self tap it on. You want the bottom of the sensor to be slightly higher than the tip of the nozzle, but not too much. 1-2mm is probably good. The goal here is for the sensor to detect the aluminum bed before the nozzle touches it.





Once the sensor is mounted you will want to remove you current Z end stop that is plugged in the Z- on the PCB, and replace it with the sensor plug. The sensor will now become your Z end stop. Remember that the sensor can only detect metal, so if it's off the side of the bed when you Z is going down, there is nothing to stop your nozzle from crashing into your bed and doing some damage to other parts.



Firmware:

If you are not familiar with how to flash your firmware and what software you need, I recommend you pause and check out his video. https://www.youtube.com/watch?v=2RbcMvhatjU

Also, make sure you are using the USB drivers in the files section of the Facebook group. https://www.facebook.com/groups/TEVO.3dprinter.owners/files/ and look for FTDI USB Drivers.zip

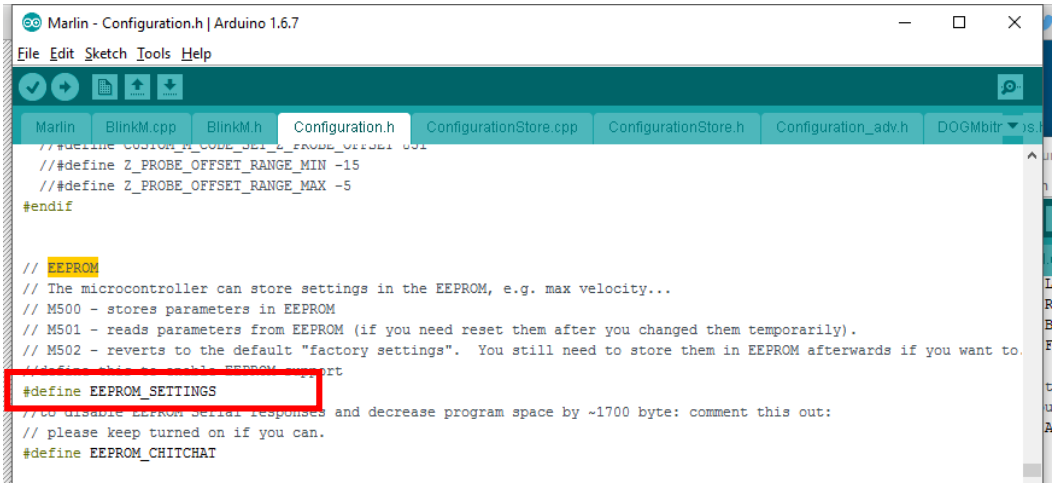
If you are running windows 8, 8.1 or 10, windows may not let you install these drivers as they are not certified. If that's the case, follow these videos to learn how to allow it. Windows update will occasionally update the drivers, so if you ever notice you can't connect to your printer from your computer anymore, that's the reason.

Windows 8 and 8.1 install non certified drivers. https://www.youtube.com/watch?v=MSCTkz3ucV8

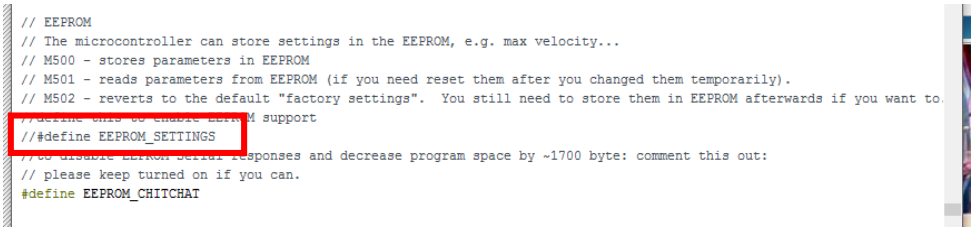
Windows 10, install non certified drivers. https://www.youtube.com/watch?v=71YAIw7_-kg

From here, I am going to be using the community firmware uploaded by Stefan. It should be titled something like tarantula-marlin.community in the files section of the Facebook group. https://www.facebook.com/groups/TEVO.3dprinter.owners/files/

After opening up the Marlin.ino file, click on the Configuration.h tab. Now, by default this firmware has EEPROM enabled. You don't want this enabled or else some settings you change in the Auto Level will not update when you upload the new firmware. Took a while to figure this out. So do a find (CTRL-F) for EEPROM and you should be here:

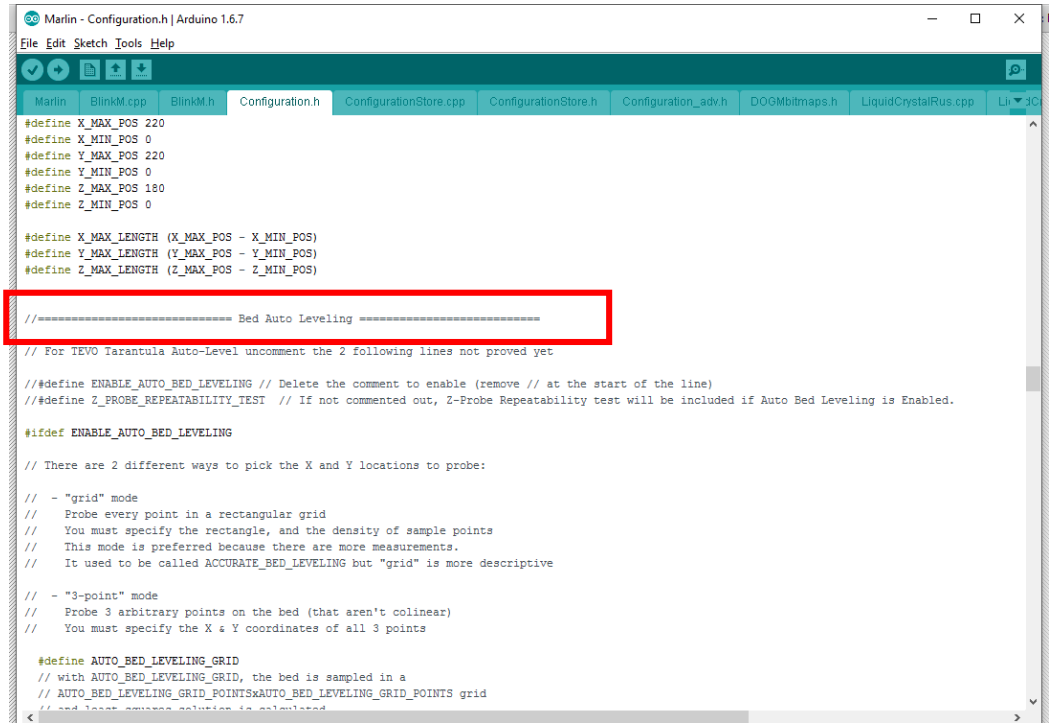


Now, you want to put two // in front of the #define EEPROM_SETTINGS.

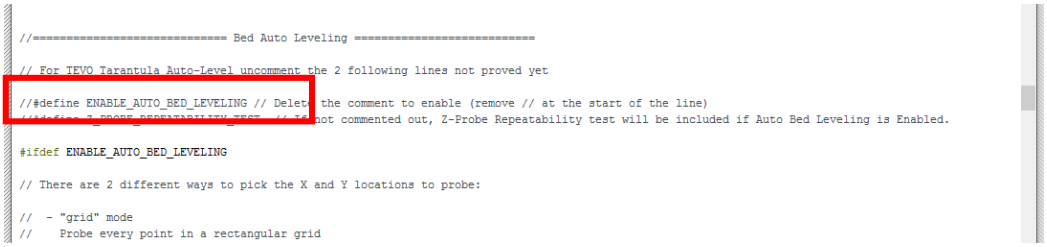


WARNING!!! If you have saved any settings in your firmware using the M500 command, those values will be overwritten by the values in this firmware now. Things like PID Autotune, E-Steps and any other calibrations you may have done. Hopefully you have those values written down and you enter them in the firmware manually. I have not taken to time to figure out which values update and which don't, so I always make changes in firmware and upload now.

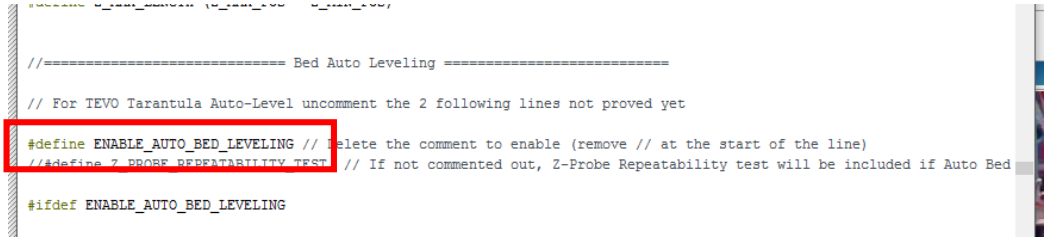
Now, scroll down until you see the Bed Auto Leveling section. You want to enable it by uncommenting (removing the //) in front of //#define ENABLE_AUTO_BED_LEVELING



This:



To this:



Scroll down a bit more until you see //set the rectangle in which to probe.

These values will tell the printer where the limits are to probe. Now depending on your home offsets, these may be different than my settings. Try mine first, but be prepared to power off the unit quickly if need be.

The second part below is telling the printer how many points in the grid. By setting it to 3, it will do a 3x3 grid (9 points). 2 will do a 2x2 grid (4 points).

```
//    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   0
    #define RIGHT_PROBE_BED_POSITION 170
    #define BACK_PROBE_BED_POSITION   180
    #define FRONT_PROBE_BED_POSITION  0

     // 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 3
```

This last part will tell the printer how your sensor is offset from your nozzle so it can make accurate calculations. The X value should be similar. My Z value might be off from yours, so you might want to start it at something like 0 and make small changes to it. This is a bit of a tedious process, but worth it once you get auto level working. You can keep the other values where I have them, or as you learn you might want to change them to fit your needs.

```
     // I wouldn't see a reason to go above 3 (=9 probing points on the bed)
    #define AUTO_BED_LEVELING_GRID_POINTS 3


  #endif // AUTO_BED_LEVELING_GRID

  // these are the offsets to the probe relative to the extruder tip (Hotend - Probe)
  #define X_PROBE_OFFSET_FROM_EXTRUDER  20
  #define Y_PROBE_OFFSET_FROM_EXTRUDER  0
  #define Z_PROBE_OFFSET_FROM_EXTRUDER -.82

  #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 XY_TRAVEL_SPEED 10000        // X and Y axis travel speed between probes, in mm/min

  #define Z_RAISE_BEFORE_PROBING 4    //How much the extruder will be raised before traveling to the first probing point.
  #define Z_RAISE_BETWEEN_PROBINGS 4  //How much the extruder will be raised when traveling from between next probing point

  //#define Z_PROBE_SLED // turn on if you have a z-probe mounted on a sled like those designed by Charles Bell
  //#define SLED_DOCKING_OFFSET 5 // the extra distance the X axis must travel to pickup the sled. 0 should be fine but you

  //If defined, the Probe servo will be turned on only during movement and then turned off to avoid jerk
```

Lastly, I would recommend you read over the Z Safe homing section. It will give you some good information on setting things up to try and avoid some crashes. I have mine set like this. I like to have the Z_SAFE_HOMING points set to the middle of the bed so I know it's in a safe spot.

```
//If you have enabled the Bed Auto Leveling and are using the same Z Probe for Z Homing,
//it is highly recommended you let this Z_SAFE_HOMING enabled!!!

  #define Z_SAFE_HOMING   // This feature is meant to avoid Z homing with probe outside the bed area.
                          // When defined, it will:
                          // - Allow Z homing only after X and Y homing AND stepper drivers still enabled
                          // - If stepper drivers timeout, it will need X and Y homing again before Z homing
                          // - Position the probe in a defined XY point before Z Homing when homing all axis (G28)
                          // - Block Z homing only when the probe is outside bed area.

  #ifdef Z_SAFE_HOMING

   // #define Z_SAFE_HOMING_X_POINT 0
    #define Z_SAFE_HOMING_X_POINT (X_MAX_LENGTH/2)    // X point for Z homing when homing all axis (G28)
    #define Z_SAFE_HOMING_Y_POINT (Y_MAX_LENGTH/2)    // Y point for Z homing when homing all axis (G28)
  #endif
```

WARNING!!!! Your HOME Offsets in firmware might have some effect on this. So just pay attention to what the following commands do. Be smart!

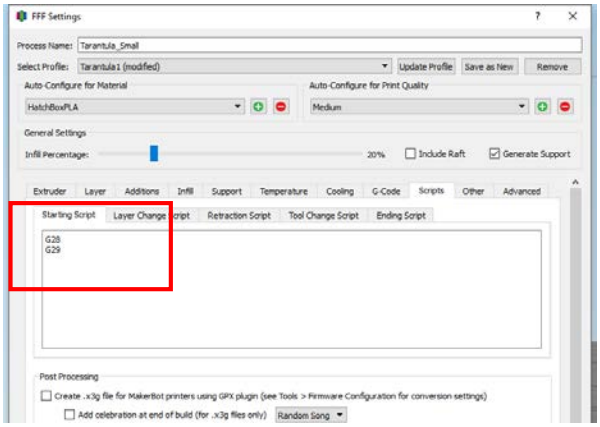
Upload the firmware to your printer and we are ready for the next step. I'm going to be using Simplify3d's printer control to do this next part, but you can use whatever your slicer software has. As long as there is a way to type in GCode commands to your printer.

Ok, with your hand on the power switch, the first thing to do is try a G28 command, this will Home all axis. Now if you did like I did with the safe homing and set Y and X points to the center, then it should home to the center of the bed now. If not, it will home to the front left corner like it always did before. Offsets might have effect on this, so pay attention.

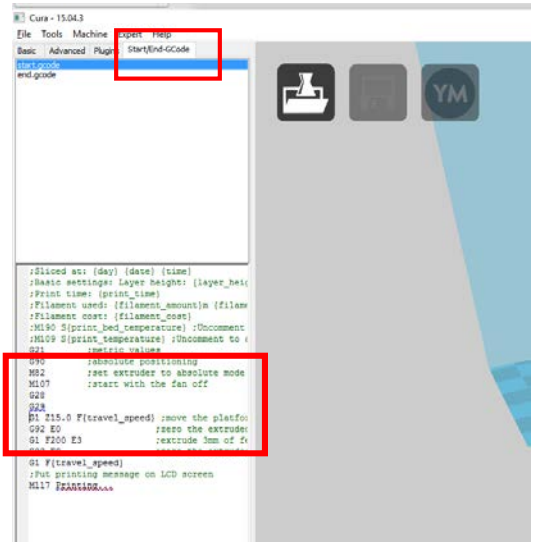
Once G28 is done and it looks safe, you can now try the G29 command. This will begin the auto level sequence. Pay close attention as this happens. If it looks like the sensor will miss the bed, power off, reset the PCB with the reset button. Make adjustments to firmware (the ***//set the rectangle in which to probe*** section), upload and try again. Always start with the G28 code first to home all. Once you are happy with all the grid point locations, you are ready to try the first print.

In your slicer there should be a place to put in starting GCode commands. Here you will put a G28, followed by a G29

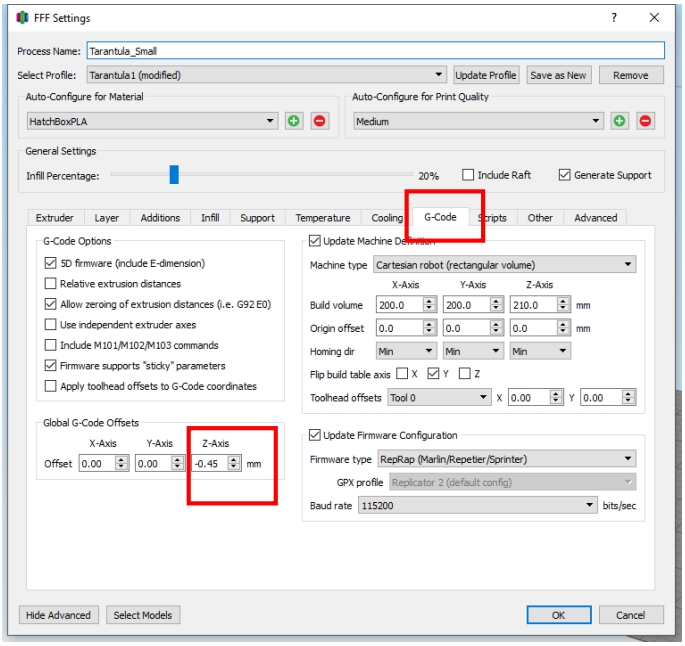
Simplify3D



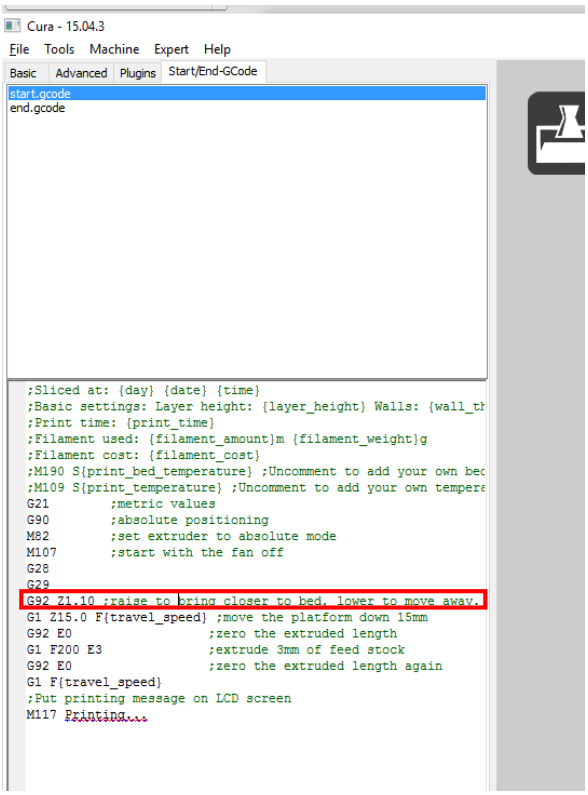
Cura



Now try a print. Obviously this first time the nozzle will probably be too high since you started with a Z probe offset of 0 in the firmware under the Auto Level section. There are a few ways to get this perfect. 1. Is changing the number in the firmware, re uploading and trying again. Slicer programs will sometimes have Z offset setting. This is nice so you don't have to keep uploading firmware to find the perfect value. In Simplify3D it's really easy:



In Cura you can set the Z offset in the start GCode section. This same idea should work for other slicers as well. The G92 is you offset command so you can do something like G92 Z0.5 to create and offset of 0.5 on the Z. You have to make sure it and put this in after the home all (G28) and auto level (G29). Play around with these values until you get just the right starting point. Like I said earlier, it's a bit tedious, but it will be worth it.



That's it. You should have all tools and knowledge you need to get going with the auto leveling feature of your Tarantula using the Marlin firmware.