How to Calibrate an Auto Bed Leveling (ABL) Sensor

Last UpdatedNovember 7, 2018 byBrett

Several months ago I wrote an in-depth article on How to Setup Auto Bed Leveling which covered the entire process from start to finish… at least almost. This was my first time using auto bed leveling and I was still testing out various methods to calibrate the sensor, where I decided to leave this for a separate guide. At that point, none of the instructions I had found worked out and print jobs would embed plastic in to the build surface, needing to be replaced several times in the process.

As I was getting tired of buying more Build Tak stickers every other week, I went looking once again and stumbled across a video from 3D Maker Noob that seemed promising. Rather than try to adjust the sensor's position by hand, he used a series of g-code commands to calibrate it with ease. Go figure, it took all of 5 minutes and my auto bed leveling sensor was near perfect.

Most of this guide will just reiterate the information he provides in the video, but several personal observations have been added as well. Once you understand it, auto bed leveling makes perfect sense, but most of the resources found online are either needlessly complicated or just plain wrong.

Software

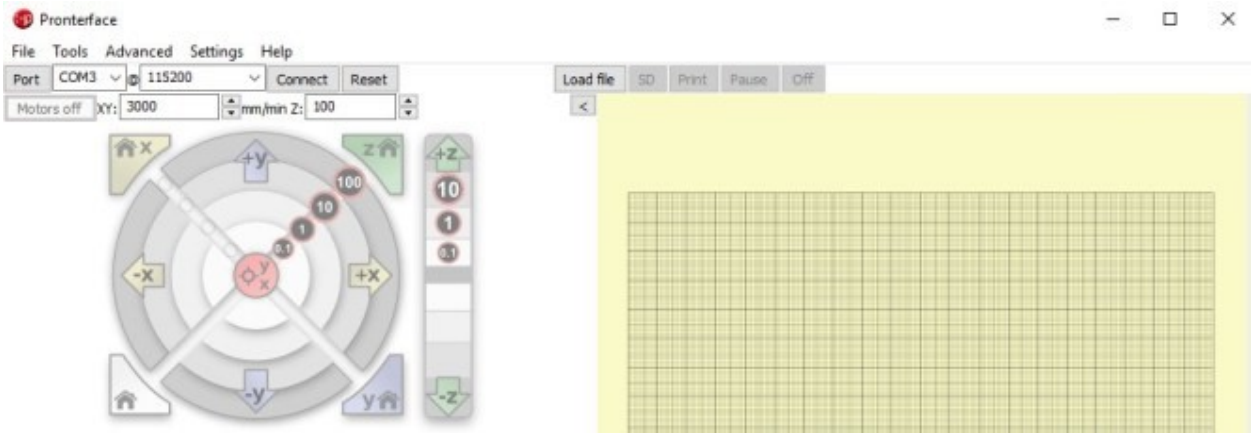
Before we can get started, we first need to grab some software that can communicate with the 3D Printer over USB. Most slicers have this feature built-in, but Cura seems to be the rare exception that lacks direct machine commands (despite otherwise being a complete printing solution). There are however several other tools I have linked below that will work just fine for our needs, where I would suggest Pronterface based on the simple, easy to use interface.

Pronterface
Repetier Host

As a third option, Simplify3D ($149) is exceptional software and my preferred slicer software of choice. It is however quite expensive and not worth the price tag for the average person, so unless you already own it, one of the free solutions above will be more than capable of getting the job done.

Calibrate Steps

Go ahead and connect the 3D Printer to your computer using a USB cable and power it on. Start up Pronterface (or whatever software you decide to use) and connect it to the machine. The Baud Rate will likely be set to the default of 115200 and the COM port will appear when the 3D Printer is connected, such as COM3.



Now our goal here is to find and set the proper Z-Offset for our nozzle, based on the position of the ABL sensor and where it triggers. The Z-Offset is an adjusted value, which raises or lowers the nozzle from the actual home position. So if we run “Auto Home” and the current nozzle position is at 0, a Z-Offset of -0.5mm would bring it closer to the bed.

It’s important to note that the Z-Offset value is ***not*** used when the printer auto homes (G28 command) or probes (G29 command) the bed, only during an actual print. This wasn’t documented anywhere and caused quite a bit of initial confusion on my part.

Step 1: Clear the Z-Offset

Since there may already be a pre-existing z-offset configured in the firmware or EEPROM, we will start by clearing out any current values and reset the offset to 0.00.

G28 Home the nozzle
M851 Z0 Set the z-offset to zero
M500 Store the settings to EEPROM
M501 Load the settings from EEPROM
M851 Echo the current z-offset value, make sure this reports Z0

Step 2: Move to Actual Z-Offset

When the Auto Bed Leveling (ABL) sensor is triggered, the firmware will raise the nozzle up by several millimeters. Since we want to work from the actual 0 z-offset, we need to re-

position it at the point the sensor was triggered. To do this, we will use *G1 F60 Z0*, where G1 is the move command, F60 is the travel speed, and Z0 is where we are moving to.

`G28 Z` Home the nozzle on the Z axis
`G1 F60 Z0` Move the nozzle down to the actual 0 offset

Step 3: Calibrate Z-Offset

Just like we are used to with manual bed leveling, go ahead and insert a piece of paper under the nozzle to test the distance. Using the movement controls in the software, start lowering the nozzle towards the build plate in increments of -0.1mm, until there is a slight drag on the paper when sliding it underneath. We will temporarily disable the software endstops, where that will allow us to go in to negative values on the Z axis while we calibrate the proper offset.

Once you are satisfied with the distance between the nozzle and the bed, make a note of the z-offset on the 3D Printer's LCD screen, which should look something like "Z-0.5". Now add the thickness of the paper you used to this value and that is your actual z-offset.

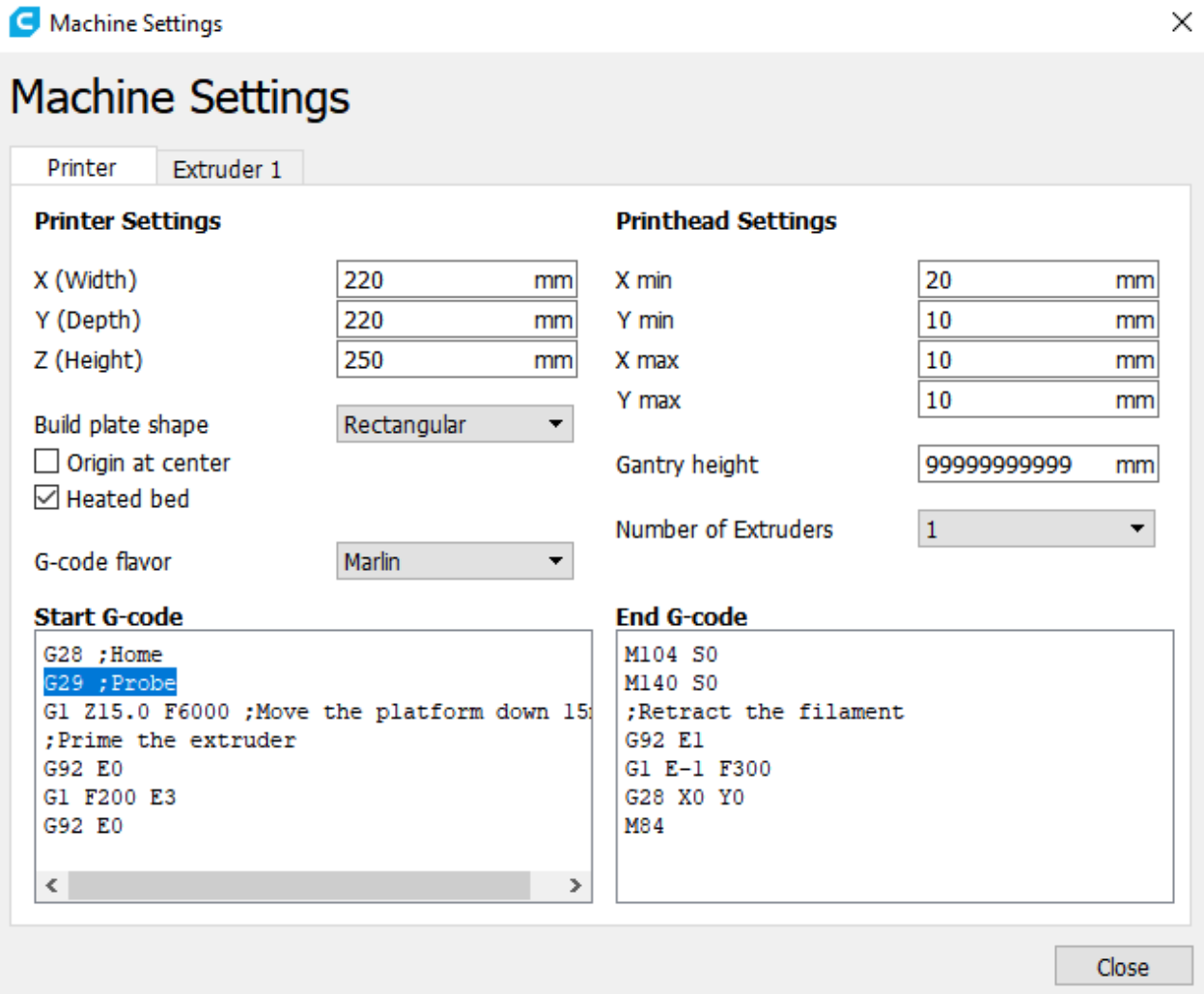
Note: *A normal sheet of paper has an average thickness of 0.08mm. If your LCD screen reads Z-0.5, we would configure this as Z-0.58.*

`M211 S0` Turn OFF the Software Endstops
`M851 Z-0.58` Set the z-offset value
`M211 S1` Turn ON the Software Endstops
`M500` Store the settings to EEPROM
`M501` Load the settings from EEPROM

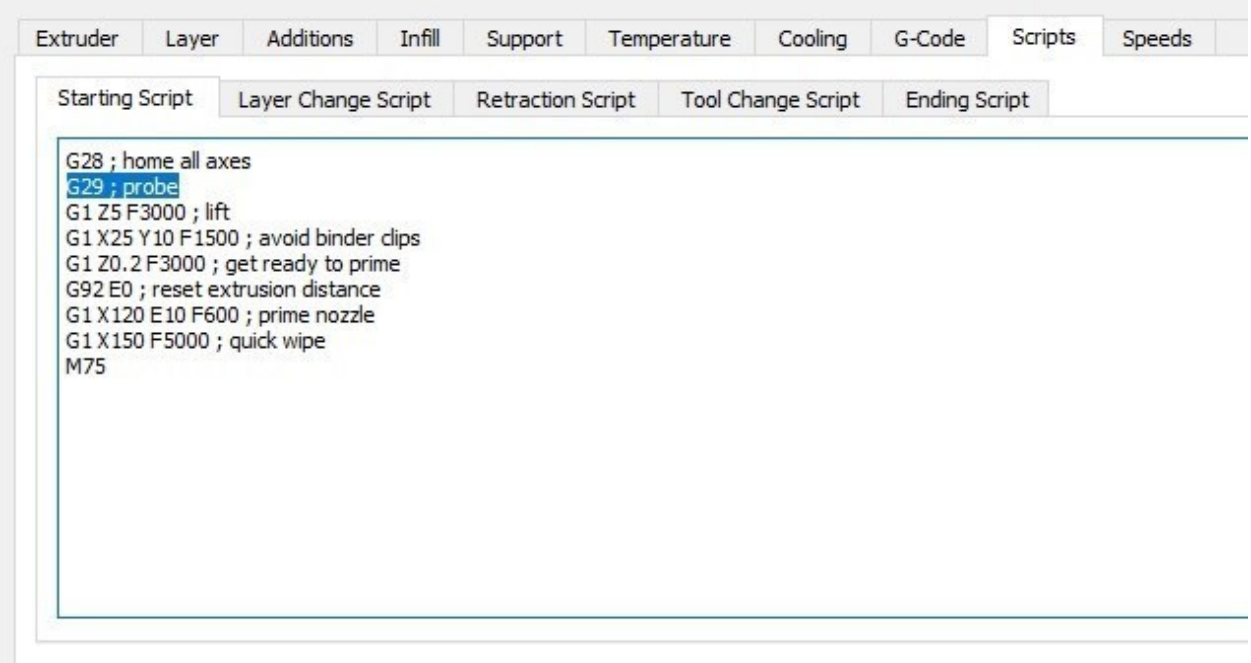
Start Script

Since we want the 3D Printer to use the auto bed leveling sensor and probe the surface before each print, we need to tell it to do so via the Start script. This is a list of G-code commands that it executes in order before the print can start. Open your preferred slicer software, locate the Start script and add the G29 probe command to it.

Cura: To access the Start script g-code in Cura, go to the Settings menu -> Printer -> Manage Printers. Select the desired printer in the list and click the *Machine Settings* button next to it. You will find the "Start G-Code" section in the bottom left.



Simplify3D: To access the Start script in Simplify3D, open the Process Settings -> Scripts tab -> Starting script tab.



Final Thoughts

As I am of the mindset “why fix what isn’t broken”, auto bed leveling always seemed like a solution for a problem that didn’t exist. Sure, it takes a few minutes to level the bed by hand, but I didn’t exactly understand what this feature was capable of. Until it was properly configured, it seemed to be a hindrance more so than anything, turning a simple process in to something complex.

Once it was setup and I eventually figured out how to calibrate the sensor, the benefits became much more apparent and it was a luxury I came to appreciate. While it of course reduces the frequency in which leveling is required, the real upside was the fact that it maps out the build surface when probing. In cases where the bed isn’t perfectly flat, it will use the generated mesh data to adapt and work around imperfections in the plate… something that manual bed leveling simply cannot do.