Upgrading the Creality Ender 5 Pro (8-bit board V1.1.5) for High-Temperature Printing

1. Buy a Creality "spider" high-temperature hotend kit. I got mine on eBay for ~$75. Other high-temperature hotends (e.g. the Phaetus Dragonfly) may work as well, but may change the procedure somewhat.

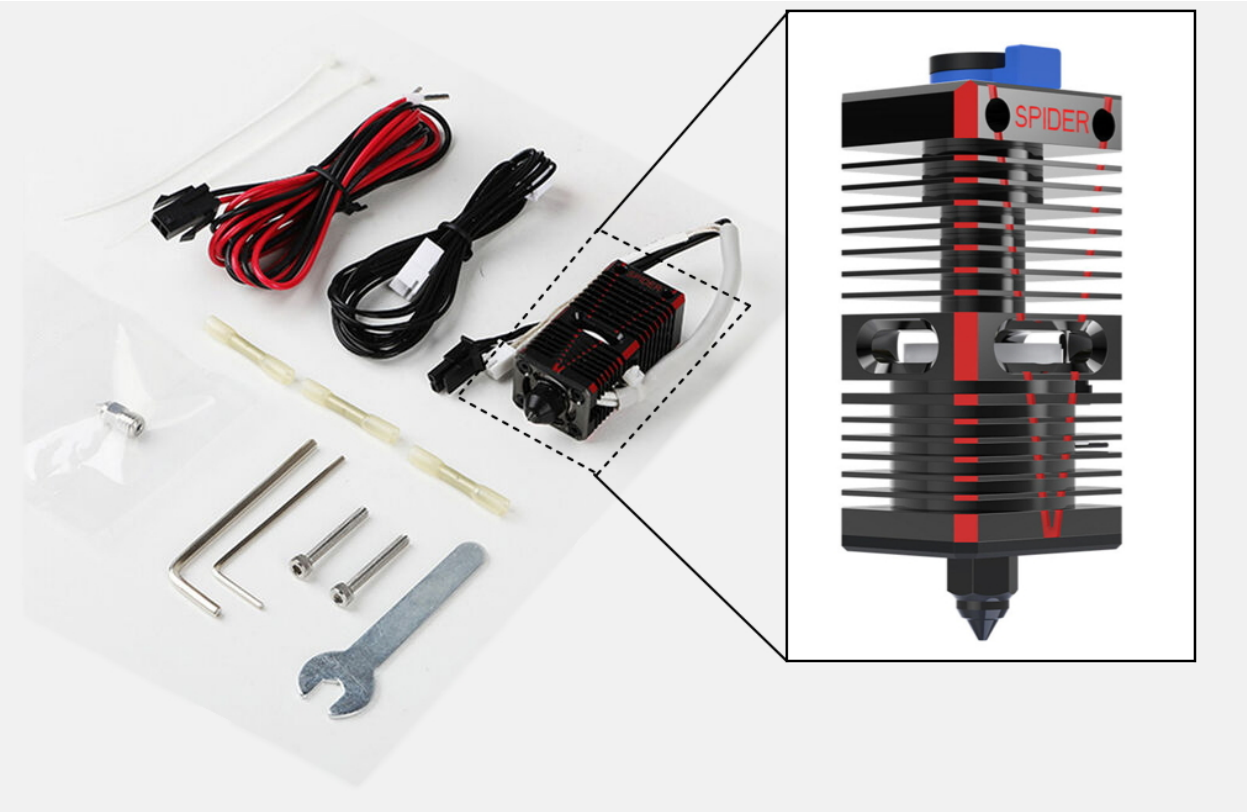

Figure 1: Creality "spider" high-temperature hotend kit.

2. Remove the old hotend and install the new one.

   To remove the stock hotend, you have to remove the cooling shroud (just take out a few screws), then remove the mounting screws that hold the stock hotend in place, and finally *cut* the wires that connect to the hotend's heating element and thermistor (temperature sensor).

   Use the mounting screws provided in the new hotend kit to mount it in place of the old one. You can't reuse the old screws because they're too short for the new hotend.

   The new hotend has short cables coming from the heater and thermistor, with plug-in connectors on them. The kit also comes with some longer cables with connectors that fit the ones on the hotend. However, the connectors on the other end of these cables don't fit anything on the printer.

   Thus, you complete the installation, you just cut the connectors that you need off the cables and solder them to existing heater and thermistor wires coming from the printer. Fortunately, the heater and thermistor are basically just resistors, so the polarity doesn't matter.

   The kit comes with some crimp-on connections that you can use for making the connections if you don't want to solder them, but for reliability, I highly recommend soldering them—I don't trust crimp-on connectors for connections like this.

3. Obtain firmware. This is basically your printer's operating system. Since the original version of the firmware only allows nozzle temperatures up to 260ºC, you need custom firmware so that you can take advantage of the higher temperature capabilities of the new hotend.

   I will include the firmware file that I ended up using with this instruction packet. However, it will probably only work for you if you have exactly the same hardware (V1.1.5 main board, same "spider" hotend, with type 104NT thermistor).

   There are various firmware versions available online, either as compiled HEX files, or as source code that you can edit and compile to create your own custom version. For the sake of convenience, I used the firmware builder utility on https://marlin.crc.id.au , which (for a small membership fee) allows you to customize, compile, and download a wide variety of firmware without needing to download source code or have your own compiler program.

   Once you have the firmware, you still have to install it on your printer. Some newer printers will check for firmware updates on the SD card and install them automatically at boot-up, so you can just put the firmware on your SD card, put it in the printer, and turn it on. However, the 8-Bit Ender 5's (mainboard versions 1.1.5 and earlier) don't have that feature. They also don't have the capability to connect to a computer via USB—they have the physical USB port, but their original firmware doesn't support communications over USB cable. Thus, if you plug the printer into a computer and try to use slicer software (e.g. Cura) to upload the new firmware to your printer, you will get an error message to the effect of "printer not connected" or "unable to establish USB connection to printer."

4. Buy a USBasp or similar programmer. This is a device that plugs into your computer and allows you to write data to a microcontroller's memory via the .

   I got my USBasp on eBay for ~$9. There are various makes and models of programmers available, but I went with the USBasp because it seems to be the most prolific. Make sure you get one with a 6-pin connection adapter, because that's what the mainboard on the printer uses.

Figure 2: USBasp programmer with cable and 6-pin adapter.

5. Install device drivers for your programmer. I will include the device drivers that I used to make the USBasp work on an old laptop running Windows 7. If you're running a different OS, you may need to download different drivers, or your computer may recognize the device automatically.
6. Install AVRDude on your computer. This is the software that allows you to upload files to microcontrollers using the USBasp. The software appears to be open-source, so you can download it for free. I will include the version of AVRDude that I used in this package, but if you're running a different OS, you may need a different version.
7. AVRDude is a command-line utility, so once you have it on your computer, you'll have to add it's directory to your computer's default search path to use it. To do this on Windows, go to Control Panel, System, Advanced System Settings, Advanced tab, and click the Environment Variables button. Then select the "path" variable, click "edit" and add the path to AVRDude. For example, C:\Program Files\AVRDude\ if that's the directory where you put the AVRDude executable file.

8. With the printer unplugged, connect the programmer to the mainboard. Because the mainboard is accessed from the bottom, and the programmer cable is rather short, I laid the printer on its side, set a piece of plywood on it, and set my laptop on the plywood, so the computer is sitting atop the printer.

   Note that you may have to unplug the LCD screen cable from the mainboard to make enough clearance around the ISP connection to plug in the programmer. Make sure you connect the 6-pin ISP header to the correct pins. The MISO pin should be labeled on the 6-pin adapter, and should be the top left pin on the mainboard header.

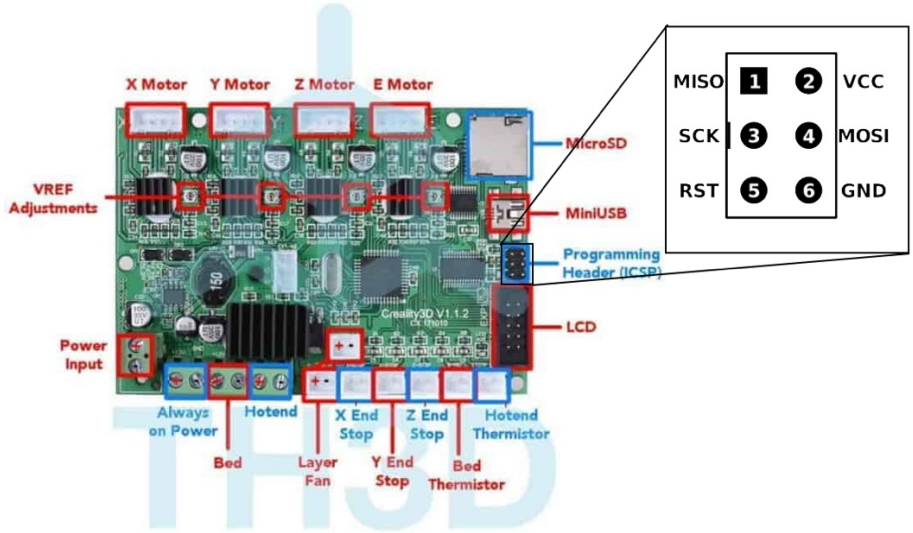

Figure 3: Pinout for connection to mainboard.

9. Upload the firmware. To do this, plug the programmer into a USB port on the computer. Then open a command prompt. You can do this on windows by typing *cmd* in the "run" box.

   In the command prompt, use the *cd* command to navigate to the folder containing your firmware file. Then enter the command:

   ```
   avrdude -c usbasp -p m1284p -u -U flash:w:hitemp.hex:i
   ```

   This command line uploads the firmware using AVRDude. The "-c usbasp" portion tells AVRDude that it's using a USBasp programmer (it can also be used with other makes and models of programmers). The "-p m1284p" portion tells AVRDude that it needs to upload the firmware to an ATMEGA1284p microcontroller, which is the chip that controls the Ender 5 Pro V1.1.5 mainboard. The "-u" specifies a safety precaution that prevents the programmer from running certain tests that could damage the microcontroller.

   Finally, "-U flash:w:hitemp.hex:i" tells AVRDude to write "hitemp.hex" to the microcontroller's flash memory in the "i" format, which is short for "intel HEX," which is the format the firmware is saved in. Note that if you're firmware file is named something other than "hitemp.hex" you'll need to change that portion of the command line to match your filename (or rename your firmware file hitemp.hex before running the command).

10. Once the process completes, disconnect the computer and programmer from the printer, set the printer back up, plug it in, and boot it up. You will probably have to tell it to "initialize the EEPROM" the first time it boots up. If it boots up correctly, it should be running the new firmware, allowing you to reach higher nozzle temperatures (up to 315ºC, if you used the same firmware I did).
11. Finally, you will probably need to double-disable the "hotend idle timeout" feature. This may just be a glitch in the firmware I used, but I suspect it is a feature that was added to the "Marlin" g-code protocol that causes older printers to malfunction because it isn't supported by the hardware.

    The feature is designed to turn off the hotend if it's above a certain temperature and the printer hasn't done anything for several minutes. However, on my printer, it sets the target nozzle temperature to zero every time the temperature rises above the threshold temperature, making it impossible to print anything, because the nozzle won't stay hot. I'm guessing this happens because the printer doesn't have an idle timer, and so it thinks it's *always* in a state of having exceeded the allowable hot-idle time.

    Thus, to fully disable the hotend idle timeout, I had to set the timeout time to zero (the normal way to disable it) *and* set the threshold temperature to the maximum nozzle temperature (315ºC), so that it never exceeds the threshold.

    Once you've made the changes, make sure to select "store settings" so the printer won't default back when you shut it down.
12. At this point, the printer should be fully functional again (with the additional capability to reach higher nozzle temperatures) but you'll probably need to re-level the bed, or at least readjust the vertical end stop to get the correct nozzle height for the first layer. At least in my case, the "spider" hotend seemed to stick out a little farther than the original factory nozzle.