

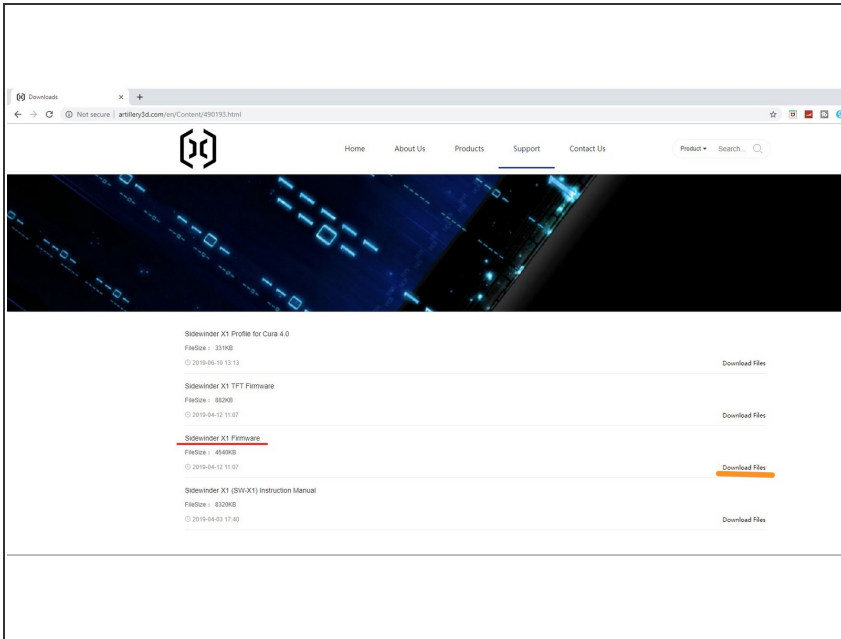


Hemera Artillery (Evnovo) Sidewinder X1 Firmware Modification

Written By: Joseph

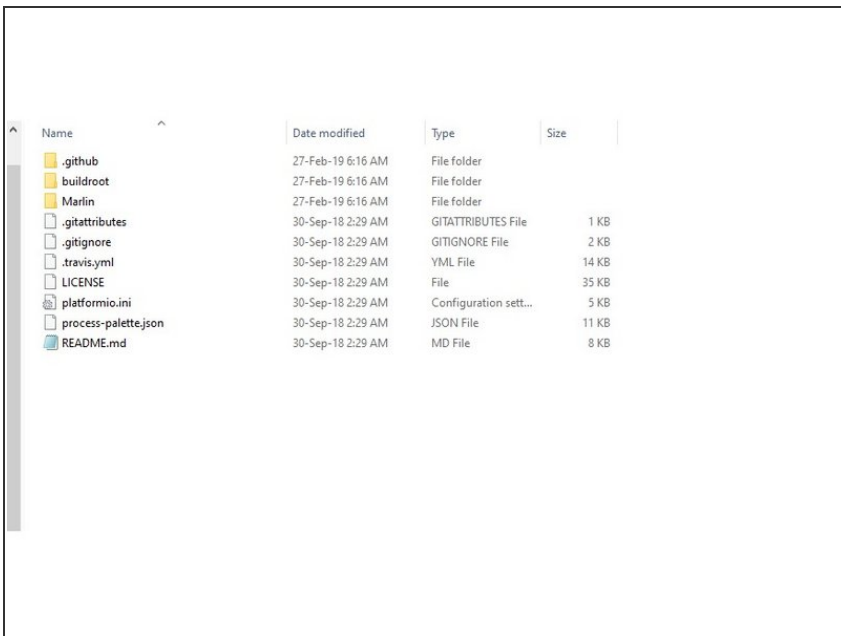


Step 1 — Firmware Download



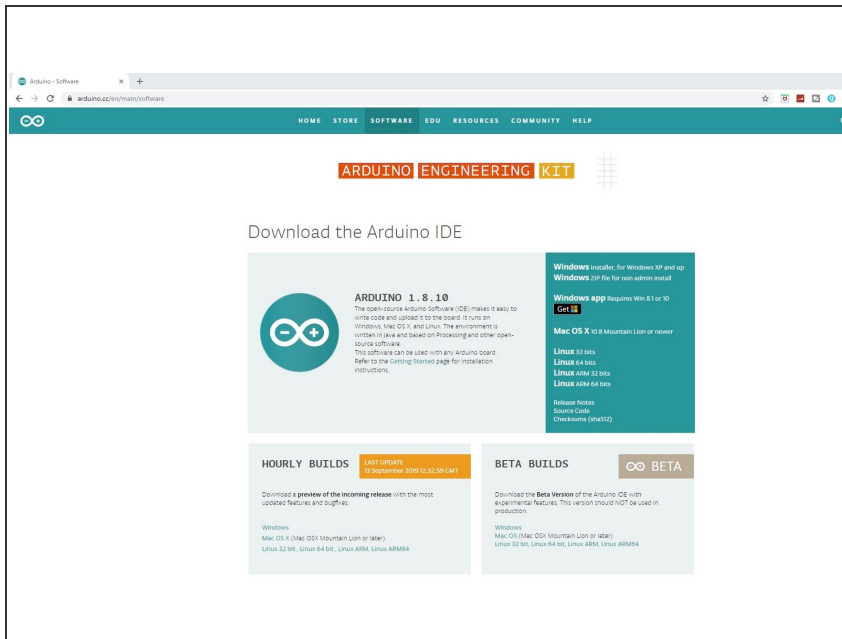
- Go to <http://artillery3d.com>
- In the support section, locate the "Sidewinder X1 Firmware"
- Click on "Download Files"

Step 2



- Once the firmware has downloaded, unzip the files

Step 3 — Arduino IDE



- If you haven't already, download the latest version of Arduino IDE from: <https://www.arduino.cc/en/main/software>

Step 4



- Open the Marlin.ino file located in the Marlin directory of the Sidewinder X1 firmware folder which you download by double-clicking on it
- This will launch Arduino IDE with the Sidewinder firmware pre-loaded
- Once loaded, select the "Configuration.h" tab

Step 5 — Temperature Sensor

```

Marlin - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Marlin Conditionals.h Conditionals_LCD.h Conditionals_post.h Configuration.h Configuration_adv.h G26_Mesh_Validation_...
* (but gives greater accuracy and more stable #I0)
* 51 : 100k thermistor - EPCOS (1k pullup)
* 52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
* 55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)
*
* 1047 : Pt1000 with 4k7 pullup
* 1010 : Pt1000 with 1k pullup (non standard)
* 147 : Pt100 with 4k7 pullup
* 110 : Pt100 with 1k pullup (non standard)
*
* Use these for Testing or Development purposes. NEVER for production machine.
* 998 : Dummy Table that ALWAYS reads 25°C or the temperature defined below.
* 999 : Dummy Table that ALWAYS reads 100°C or the temperature defined below.
*
* : { '0': "Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC Semitec 204GT-2", '3':"Mendel-parts / 4.7k", '4'
*/
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_BED 1
#define TEMP_SENSOR_CHAMBER 0
// Dummy thermistor constant temperature readings, for use with 998 and 999
#define DUMMY_THERMISTOR_998_VALUE 25
<
Marlin - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Marlin Conditionals.h Conditionals_LCD.h Conditionals_post.h Configuration.h Configuration_adv.h G26_Mesh_Validation_...
* (but gives greater accuracy and more stable #I0)
* 51 : 100k thermistor - EPCOS (1k pullup)
* 52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
* 55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)
*
* 1047 : Pt1000 with 4k7 pullup
* 1010 : Pt1000 with 1k pullup (non standard)
* 147 : Pt100 with 4k7 pullup
* 110 : Pt100 with 1k pullup (non standard)
*
* Use these for Testing or Development purposes. NEVER for production machine.
* 998 : Dummy Table that ALWAYS reads 25°C or the temperature defined below.
* 999 : Dummy Table that ALWAYS reads 100°C or the temperature defined below.
*
* : { '0': "Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC Semitec 204GT-2", '3':"Mendel-parts / 4.7k", '4'
*/
#define TEMP_SENSOR_0 5
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_BED 1
#define TEMP_SENSOR_CHAMBER 0
// Dummy thermistor constant temperature readings, for use with 998 and 999
#define DUMMY_THERMISTOR_998_VALUE 25
<

```

- Scroll down until you find the Thermal settings section
- You should see a line that reads `#define TEMP_SENSOR_0 1`
- Change the last number of that line to 5 instead of 1. It should read `"#define TEMP_SENSOR_0 5"`
- ⓘ This means you are changing the temperature sensor type to a 100K thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (4.7k pullup). If you skip this step the temperature will not be read correctly

Step 6 — Maximum Temperature

```

win - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
dt Sketch Tools Help

line TEMP_HYSTERESIS 5 // (degC) range of +/- temperatures considered "close" to the target one
line TEMP_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

ed temperature must be close to target for this long before M190 returns success
line TEMP_RESIDENCY_TIME 10 // (seconds)
line TEMP_HYSTERESIS 3 // (degC) range of +/- temperatures considered "close" to the target one
line TEMP_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

he minimal temperature defines the temperature below which the heater will not be enabled It is used
o check that the wiring to the thermistor is not broken.
hewise this would lead to the heater being powered on all the time.
ne HEATER_0_MAXTEMP 5
line HEATER_1_MAXTEMP 5
line HEATER_2_MAXTEMP 5
line HEATER_3_MAXTEMP 5
line HEATER_4_MAXTEMP 5
line BED_MAXTEMP 5

hen temperature exceeds max temp, your heater will be switched off.
is feature exists to protect your hotend from overheating accidentally, but "NOT" from thermistor short/failure!
ou should use MINTEMP for thermistor short/failure protection.
ne HEATER_0_MAXTEMP 275
line HEATER_1_MAXTEMP 275
line HEATER_2_MAXTEMP 275
line HEATER_3_MAXTEMP 275
line HEATER_4_MAXTEMP 275
line BED_MAXTEMP 150

===== PID Settings =====
ID Tuning Guide here: https://reprap.org/wiki/PID\_Tuning

omment the following line to disable PID and enable bang-bang.
line PIDTEMP
line BANG_MAX 255 // Limits current to nozzle while in bang-bang mode; 255=full current
line PID_MIN_BANG_MAX // Limits current to nozzle while PID is active (see PID_FUNCTIONAL_RANGE below); 255=full current
line PID_M 0.95 // Smoothing factor within any PID loop

```

```

win - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
dt Sketch Tools Help

line TEMP_HYSTERESIS 5 // (degC) range of +/- temperatures considered "close" to the target one
line TEMP_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

ed temperature must be close to target for this long before M190 returns success
line TEMP_RESIDENCY_TIME 10 // (seconds)
line TEMP_HYSTERESIS 3 // (degC) range of +/- temperatures considered "close" to the target one
line TEMP_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

e minimal temperature defines the temperature below which the heater will not be enabled It is used
o check that the wiring to the thermistor is not broken.
hewise this would lead to the heater being powered on all the time.
ne HEATER_0_MAXTEMP 5
line HEATER_1_MAXTEMP 5
line HEATER_2_MAXTEMP 5
line HEATER_3_MAXTEMP 5
line HEATER_4_MAXTEMP 5
line BED_MAXTEMP 5

en temperature exceeds max temp, your heater will be switched off.
is feature exists to protect your hotend from overheating accidentally, but "NOT" from thermistor short/failure!
ou should use MINTEMP for thermistor short/failure protection.
ne HEATER_0_MAXTEMP 300
line HEATER_1_MAXTEMP 275
line HEATER_2_MAXTEMP 275
line HEATER_3_MAXTEMP 275
line HEATER_4_MAXTEMP 275
line BED_MAXTEMP 150

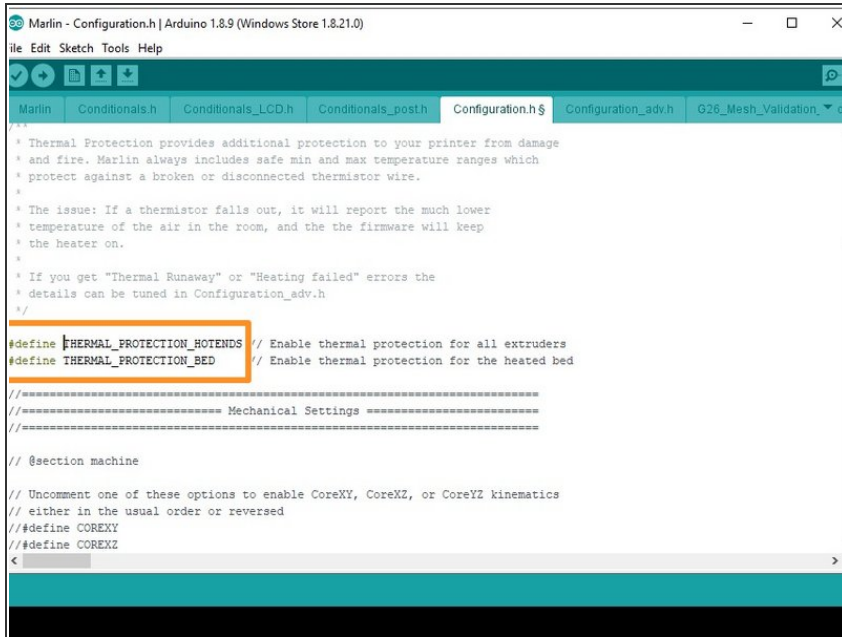
===== PID Settings =====
ID Tuning Guide here: https://reprap.org/wiki/PID\_Tuning

omment the following line to disable PID and enable bang-bang.
ne PIDTEMP
line BANG_MAX 255 // Limits current to nozzle while in bang-bang mode; 255=full current
line PID_MIN_BANG_MAX // Limits current to nozzle while PID is active (see PID_FUNCTIONAL_RANGE below); 255=full current
line PID_M 0.95 // Smoothing factor within any PID loop

```

- Scroll down a bit further and you will find the minimum and maximum temperate settings
- You should find a line that read "#define HEATER_0_MAXTEMP 275"
- Change the last 3 digits to 300 - "#define HEATER_0_MAXTEMP 300"
- ⓘ Marlin has a -15° restriction on the maximum temperature able to be set on the LCD, so in order to hot tighten at 285°C you will need to set the max temperature to 300°C. Never exceed 285°C when using the thermistor with Hermes

Step 7 — Thermal Runaway



```
Marlin - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
file Edit Sketch Tools Help

Marlin | Conditionals.h | Conditionals_LCD.h | Conditionals_post.h | Configuration.h $ | Configuration_adv.h | G26_Mesh_Validation


*/
 * Thermal Protection provides additional protection to your printer from damage
 * and fire. Marlin always includes safe min and max temperature ranges which
 * protect against a broken or disconnected thermistor wire.
 *
 * The issue: If a thermistor falls out, it will report the much lower
 * temperature of the air in the room, and the the firmware will keep
 * the heater on.
 *
 * If you get "Thermal Runaway" or "Heating failed" errors the
 * details can be tuned in Configuration_adv.h
 */

#define THERMAL_PROTECTION_HOTENDS // Enable thermal protection for all extruders
#define THERMAL_PROTECTION_BED // Enable thermal protection for the heated bed

//===== Mechanical Settings =====
//=====

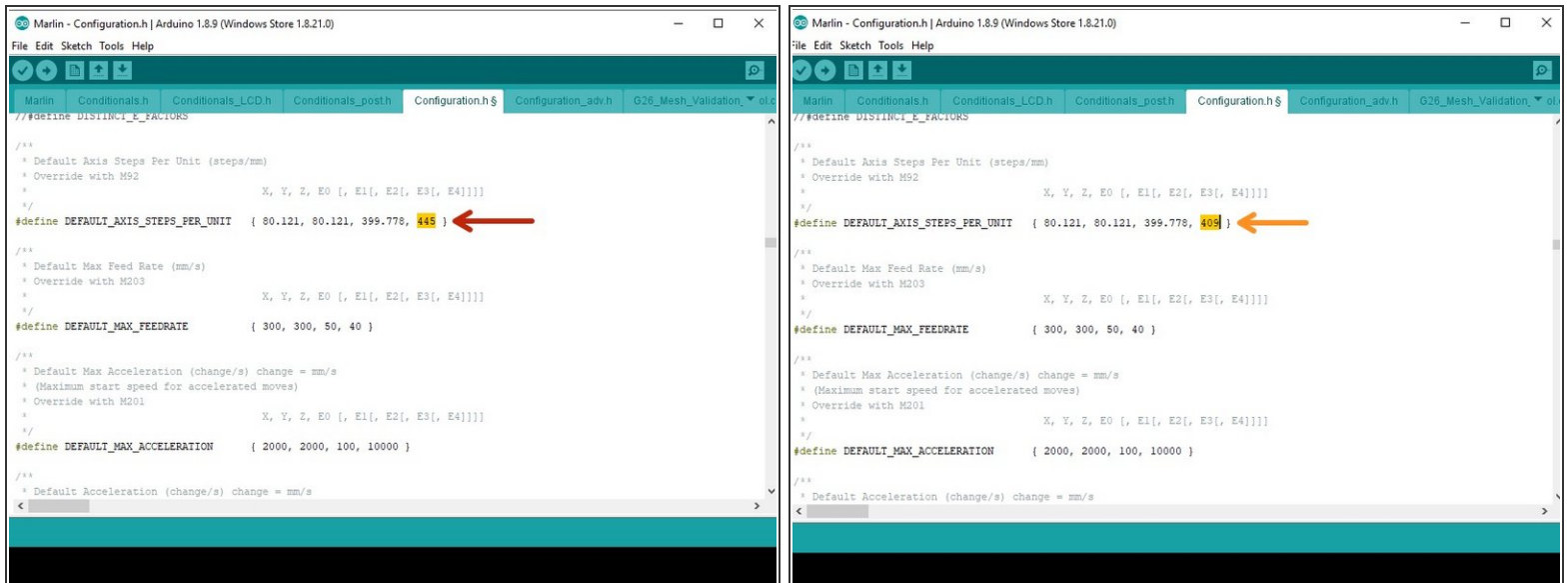
// @section machine

// Uncomment one of these options to enable CoreXY, CoreXZ, or CoreYZ kinematics
// either in the usual order or reversed
//#define COREXY
//#define COREXZ
```

 Whenever modifying any firmware, always make sure that Thermal Runaway is enabled

- Scroll down until you find the line that reads "#define THERMAL_PROTECTION_HOTENDS"
- If that line, and the one underneath it have "/" in the beginning of the string, remove them to enable Thermal Runaway protection

Step 8 — E-Steps



```
Marlin - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Marlin Conditionals.h Conditionals_LCD.h Conditionals_post.h Configuration.h$ Configuration_adv.h G26_Mesh_Validation...
// #define DISTINCT_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80.121, 80.121, 399.778, 445 }

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_MAX_FEEDRATE { 300, 300, 50, 40 }

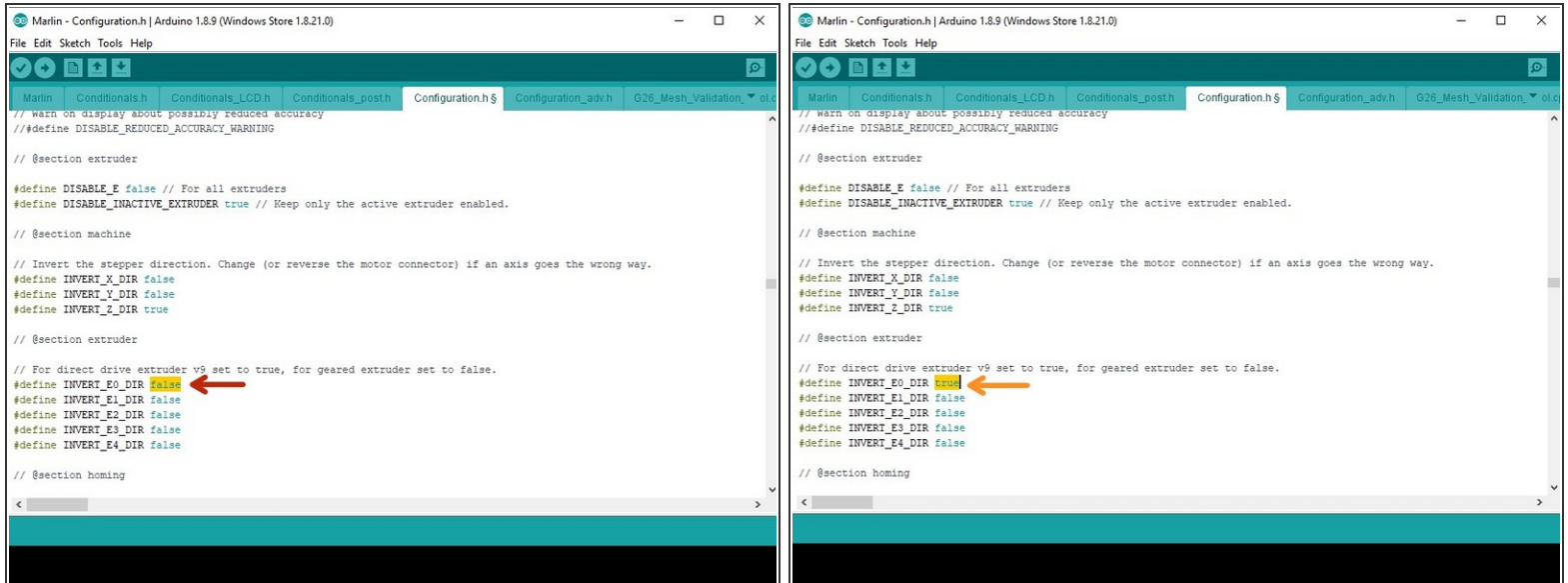
/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_MAX_ACCELERATION { 2000, 2000, 100, 10000 }

/**
 * Default Acceleration (change/s) change = mm/s

```

- Scroll down to the line that reads "#define DEFAULT_AXIS_STEPS_PER_UNIT { 80.121, 80.121, 399.778, 445 }"
- Change the last 3 digits to 409 in order to set the correct steps for the Hermes extruder
- The line should now read: "#define DEFAULT_AXIS_STEPS_PER_UNIT { 80.121, 80.121, 399.778, 409 }"

Step 9 — Stepper Direction



```
Marlin - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Marlin Conditionals.h Conditionals_LCD.h Conditionals_post.h Configuration.h Configuration_adv.h G26_Mesh_Validation.o.c
// Warn on display about possibly reduced accuracy
// #define DISABLE_REDUCE_ACCURACY_WARNING

// @section extruder

#define DISABLE_E false // For all extruders
#define DISABLE_INACTIVE_EXTRUDER true // Keep only the active extruder enabled.

// @section machine

// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR false
#define INVERT_Y_DIR false
#define INVERT_Z_DIR true

// @section extruder

// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false

// @section homing

Marlin - Configuration.h | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Marlin Conditionals.h Conditionals_LCD.h Conditionals_post.h Configuration.h Configuration_adv.h G26_Mesh_Validation.o.c
// Warn on display about possibly reduced accuracy
// #define DISABLE_REDUCE_ACCURACY_WARNING

// @section extruder

#define DISABLE_E false // For all extruders
#define DISABLE_INACTIVE_EXTRUDER true // Keep only the active extruder enabled.


// @section machine

// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR false
#define INVERT_Y_DIR false
#define INVERT_Z_DIR true

// @section extruder

// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR true
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false

// @section homing
```

 Since the Hemera is geared, it translates into the final gear rotating the opposite direction from the stock extruder, so the direction needs to be inverted

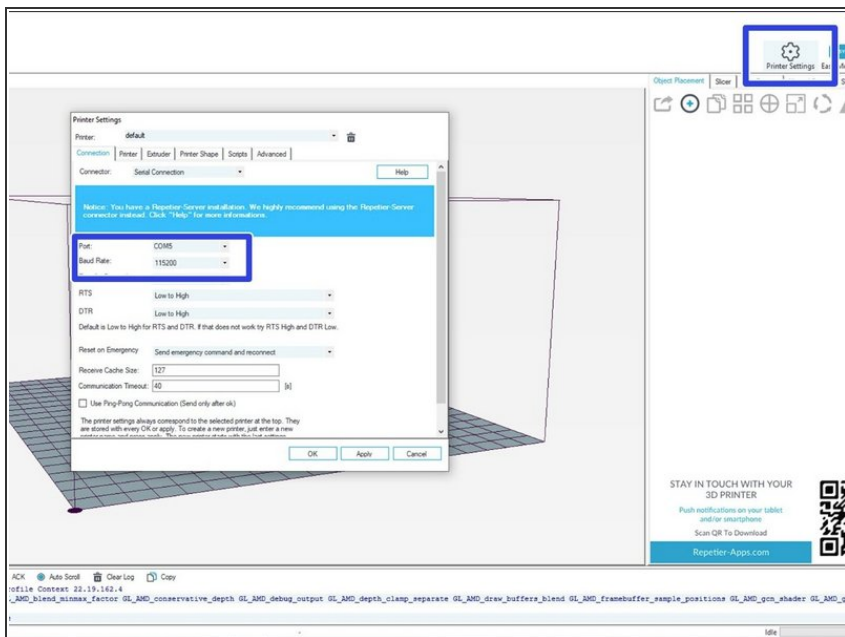
- Locate the line that reads "#define INVERT_E0_DIR false"
- Change the word "false" to "true" using only lowercase letters
- The line should now read "#define INVERT_E0_DIR true"

Step 10



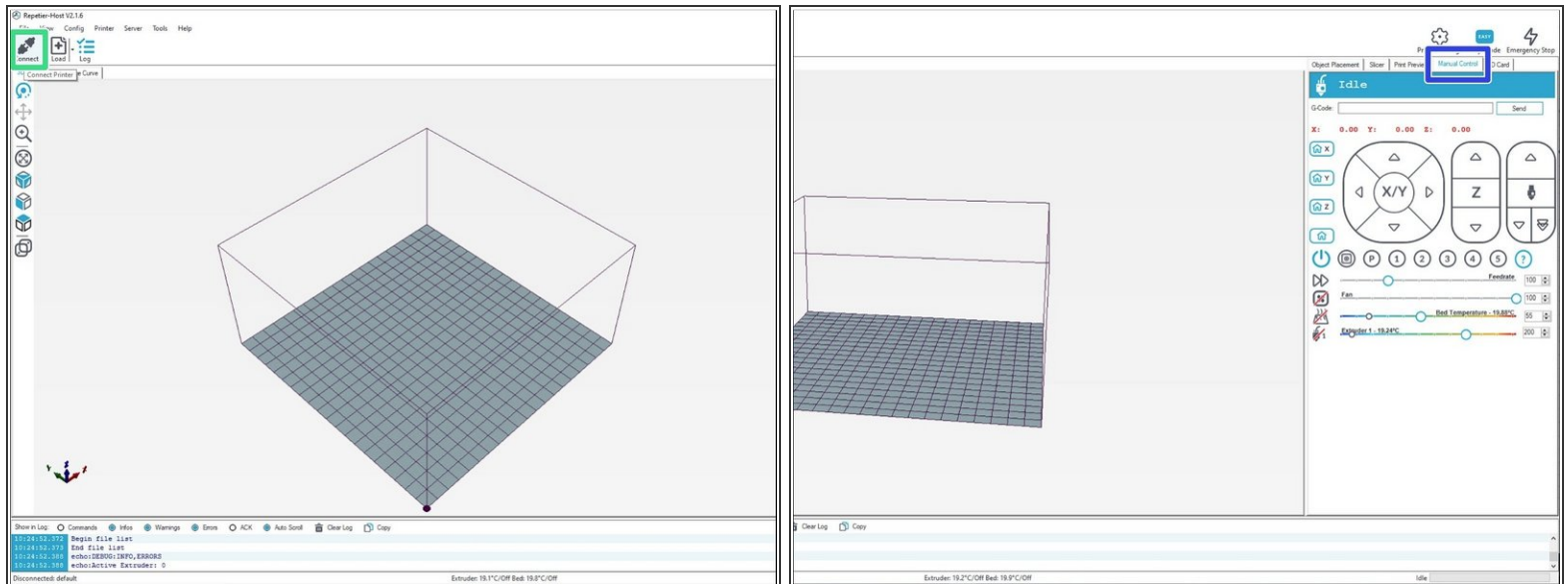
- In order to make sure the temperature reading is as stable as possible it is important to PID tune the setup.
 - The easiest way to do this is to use Repetier Host.
 - Repetier Host can be downloaded here: <https://www.repetier.com/>
- ⚠ Only run a PID tune after you have uploaded the new thermal settings to the Sidewinder X1.

Step 11



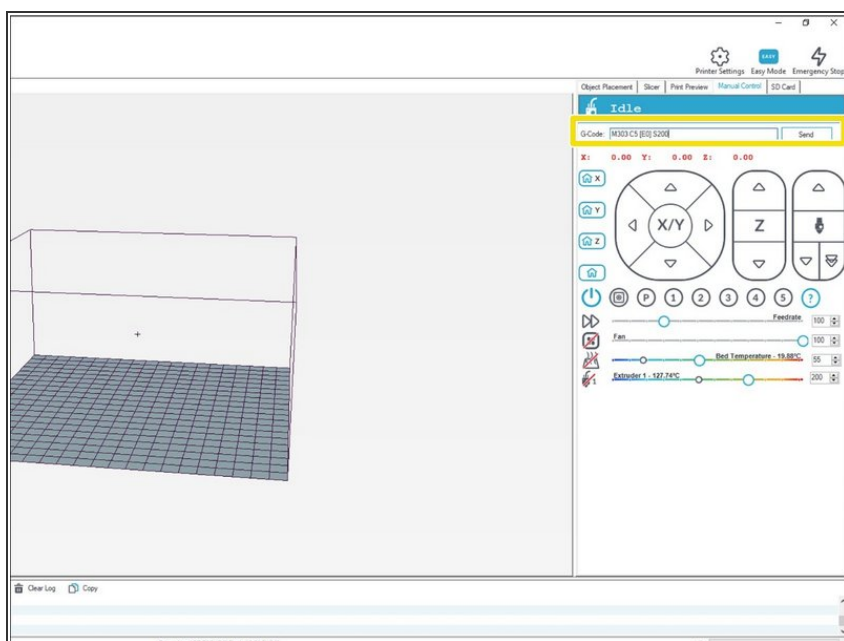
- Once you have downloaded Repetier Host open it up and click printer settings
- Make sure the COM port you have the printer plugged into is selected.
- Make sure the Baudrate is 115200

Step 12



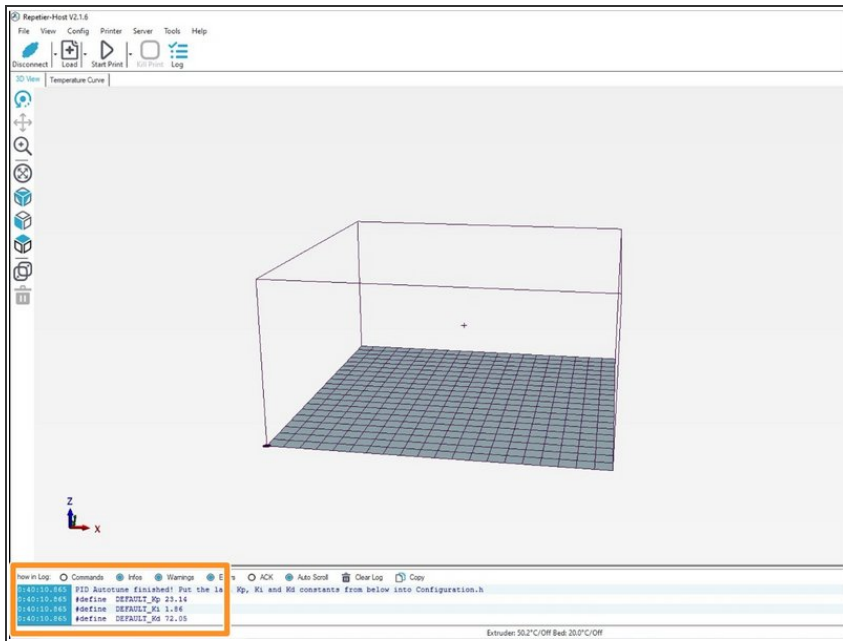
- Select connect
- Select manual control
- This will then allow the entry of Gcode

Step 13



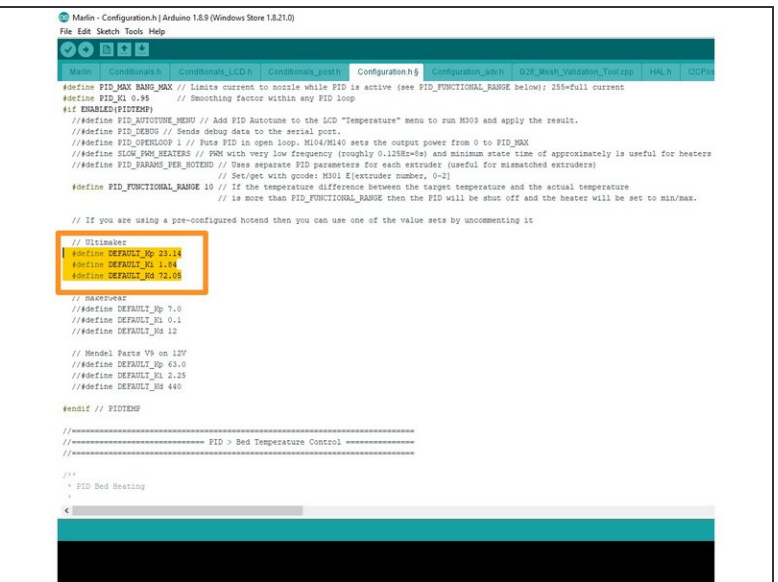
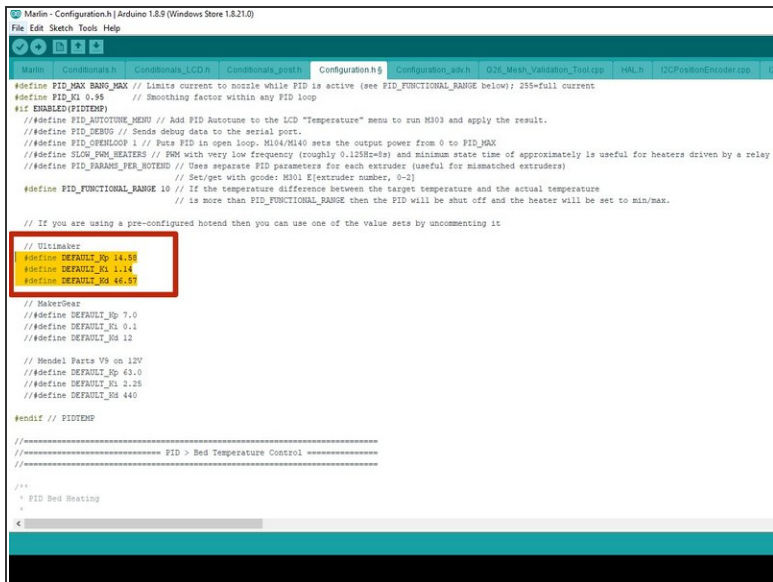
- Enter "M303 C5 [E0] S200" into the terminal
- Hit enter

Step 14



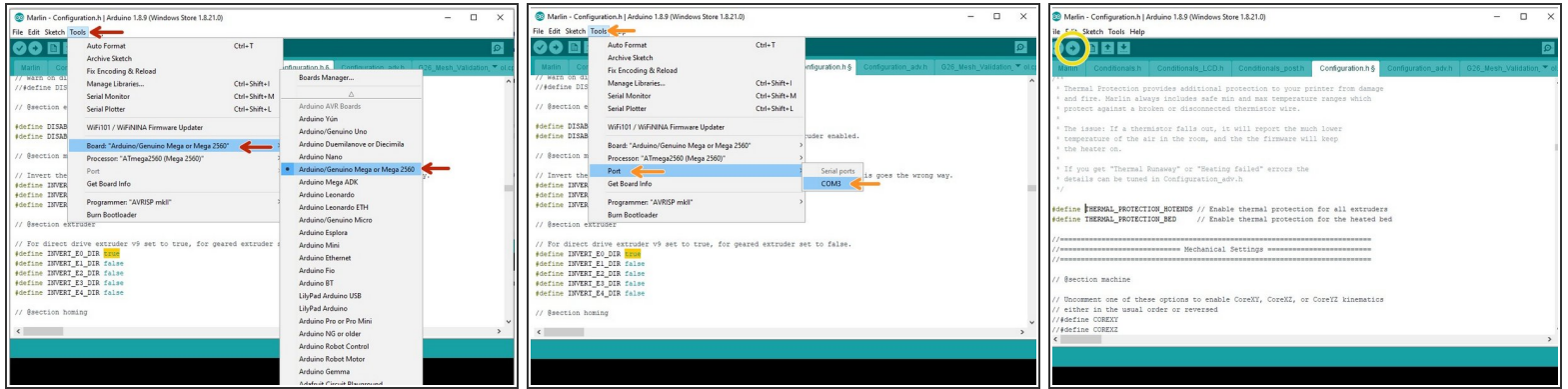
- After it has run through the 5 cycles that were set it will show the Kp, Ki, and Kd values in the dialog box.
- Make a note of these.

Step 15



- Open the Arduino IDE again and find the PID settings in configuration.h
- Replace the old values with the ones you just gained from the PID tune.
- The values entered here will not necessarily be the same as yours.

Step 16



- To upload the firmware, select the correct board as shown
- Select the Correct COM port, this may be a different COM port number
- Click on upload