CoreXY CNC Plotter

by lingib

This instructable describes an A2 plotter made from low cost aluminium extrusion, a piece of particle board, two NEMA17 stepper motors, and a few belts and pulleys.

The plotter is:

- high resolution (80 steps/mm),
- fast,
- low-cost,
- and scaleable

The plotter has an on-board interpreter that recognizes the g-code output from "Inkscape" .

Metal work is simple ... all you need is a hacksaw, three drills, a rat-tail file, and a screwdriver.

12 October 2017:

An improved pen-lift is described in instructable https://www.instructables.com/id/CNC-Pen-Lift/

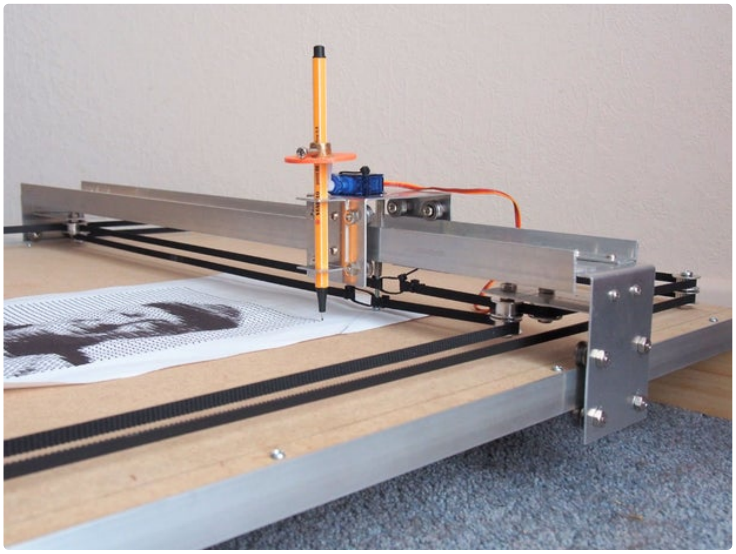
16 November 2017:

Bluetooth is added in instructable https://www.instructables.com/id/Add-Bluetooth-to-...

Z-axis control is added in instuctable https://www.instructables.com/id/CNC-Pen-and-Wash-...

17 March 2018:

GRBL (optional) installed in instructable https://www.instructables.com/id/How-to-Control-a-...



https://youtu.be/YUxcEV3zvNY

Step 1: Circuit

The wiring diagram for this plotter is shown in photo 1.

The EasyDriver module expects the wires from each motor coil to be adjacent. Check that the motor wires don't alternate ... if so swap the two center wires.

Adjusting the motor current(s):

Set your CPS-3205 power supply to 12 volts.

Attach a 12 volt 30 ohm NEMA17 stepping motor to a Big EasyDriver module and apply power to the motor assemby.

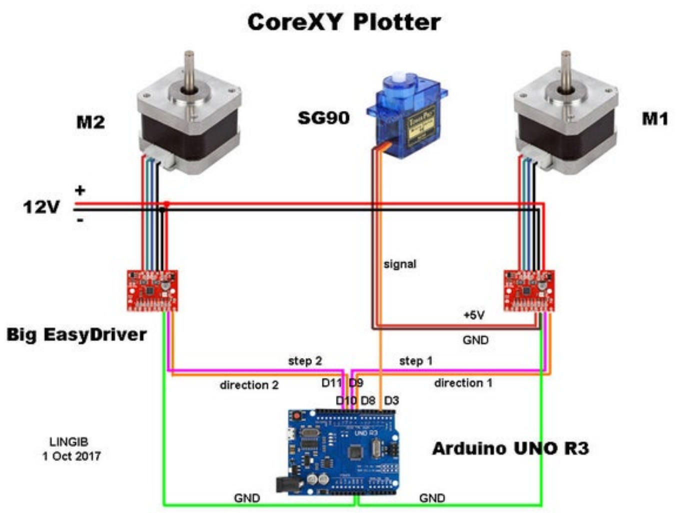
Now adjust the small potentiometer on the Big EasyDriver module for a current reading of 0.4 amps (400mA) on the CPS-3205 power supply.

Repeat this process for the remaining motor and Big EasyDriver module.

Disconnect the power.

Substitute motors:

The plotter also works with 6 volt 8 ohm stepping motors in which case the CPS-3205 power supply should be set to 6 volts and the motor currents set to 0.6 amps (600mA).



Step 2: Software

The source code for this plotter is attached.

Software installation:

- copy the contents of coreXY_plotter.ino into an arduino sketch. **[1]**
- save the file as "coreXY_plotter" (without the quotes).
- compile and upload the sketch to your arduino.

[1]

22 December 2018:

The "bool" definitions in the file coreXY_plotter_1a.ino have been rewritten in the following format:

- bool CW = true; //flag ... does not affect motor direction
- bool CCW = false; //flag ... does not affect motor direction
- bool DIRECTION1; //motor directions can be changed in step_motors()
- bool DIRECTION2;

For some reason the original format (in some compilers) no longer works ????

Otherwise the code has not been altered.

https://www.instructabl... Download

https://www.instructabl... Download

Step 3: Parts List

The parts list for either an A4 or A2 coreXY plotter is attached ...

Wood saws are not required if you ask your timber merchant to cut the particle board and and pine supports to length.

https://www.instructabl... Download

Step 4: Theory

The timing belts are arranged in a "coreXY" configuration.

An excellent explanation of the "coreXY" principle may be found at http://corexy.com/theory.html

The pen moves:

- horizontally when the motors rotate in the same direction.
- vertically when the motors rotate in opposite directions.
- diagonally if only one motor rotates.

Step 5: The Mathematics

This step derives the pen motion equations and may be ignored ...

If we rotate motor A counter-clockwise by an amount ΔA then the pen carriage will move horizontally to the right by an amount ΔX and vertically upwards by an amount ΔY. The equation for this is:

$\Delta A = \Delta X + \Delta Y$ .... (1)

If we rotate motor B counter-clockwise by an amount ΔB then the pen carriage will move horizontally to the right by an amount ΔX and vertically down by an amount ΔY. The equation for this is:

$\Delta B = \Delta X - \Delta Y$ .... (2)

The reason for the ΔY sign/direction change is that ***the two timing-belts are moving in opposite directions***... the timing-belt for motor A is being fed towards the pen which lets the pen to move upwards, whereas the timing-belt for motor B drags the pen downwards.

Horizontal motion:

Rearranging equations (1) and (2) we get:

$\Delta Y = \Delta A - \Delta X$ .... (3)

$\Delta Y = -\Delta B + \Delta X$ .... (4)

Equating equations (3) and (4) we get:

$\Delta A - \Delta X = -\Delta B + \Delta X$ .... (5)

From which:

$\Delta X = (\Delta A + \Delta B)/2$ .... (6)

Translating: the pen moves horizontally when both motors rotate in the same direction

Vertical motion:

Rearranging equations (1) and (2) we get:

$\Delta X = \Delta A + \Delta Y$ .... (7)

$\Delta X = \Delta B - \Delta Y$ .... (8)

Equating equations (7) and (8) we get:

$\Delta A + \Delta Y = \Delta B - \Delta Y$ .... (9)

From which:

$\Delta Y = (\Delta A - \Delta B)/2$ ....(10)

Translating: the pen moves vertically when the motors to rotate in opposite directions.

Collectively equations (6) and (10) infer diagonal movement if only one motor rotates.

Step 6: Evolution ... a Story of Trial and Error

H-Bot design:

My first build comprised a single timing belt arranged in the shape of a letter 'H', or "H-Bot" configuration, as shown in photo1.

I quickly abandoned this design for three reasons:

- If both motors rotate in the same direction the tension at opposite ends of the gantry caused unacceptable "wracking" [1].
- pen wobble was excessive
- the timing-belt tension spring was not a good idea ... the belt tension needs to be firm.

CoreXY design:

A few modfications eliminated all of the above problems.

- The timing belt was cut in half and arranged in a "coreXY" configuration [2]. This arrangement has the advantage that all unbalanced timing-belt tension is in the direction of pen movement as shown by the arrows in photo 2 and photo 3.
- Pen wobble was eliminated by using an extra guide rail for the pen assembly.
- Variations in belt tension were eliminated by replacing the spring with cable-ties.

The design is scalable:

Photo 4 compares the original A4 plotter with the base of a larger A2 plotter described in step 7.

The gantry and pen carriage dimensions are common to all plotters. The only additional parts required for the larger plotter shown in photo 5 are:

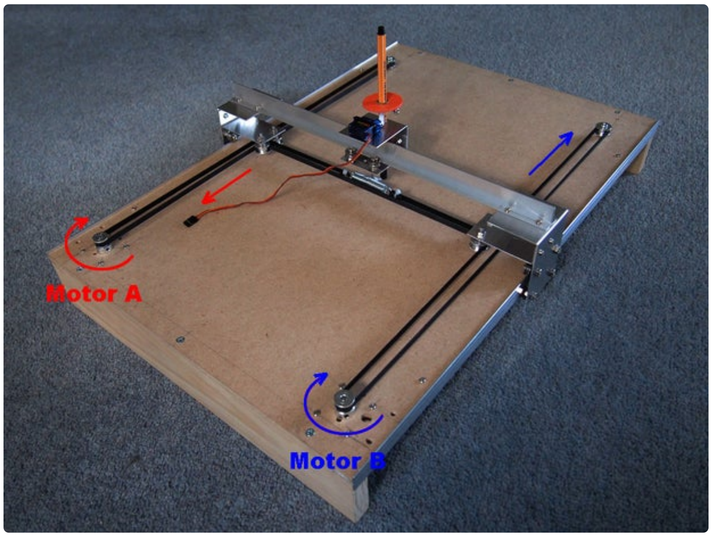
- a larger base board.
- longer side rails.
- longer timing belts

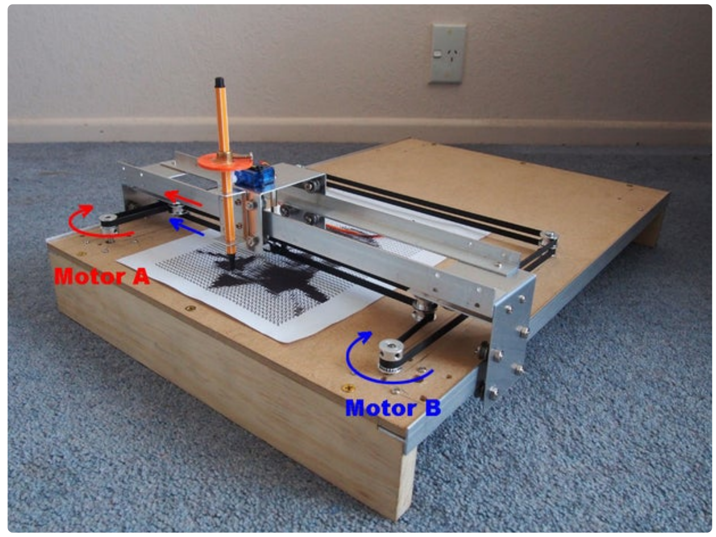
[1]

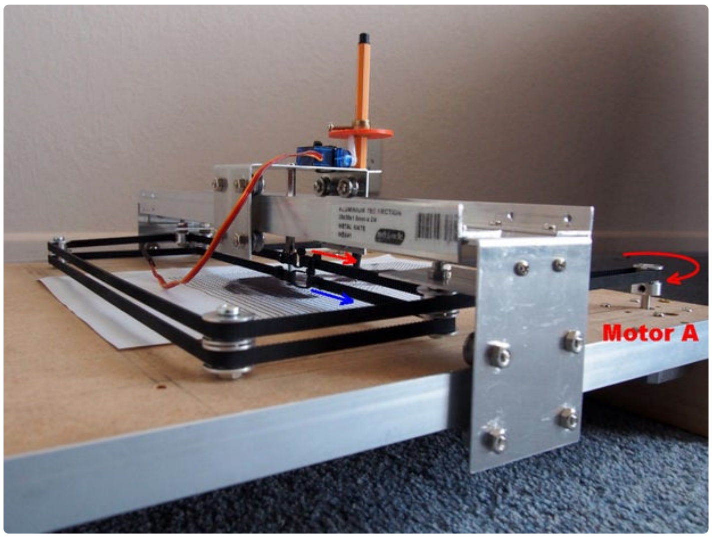
An H-Bot timing belt pulls the gantry ends in opposite directions whenever the pen moves horizontally (see arrow directions in photo 1). This causes a twisting motion known as "wracking".

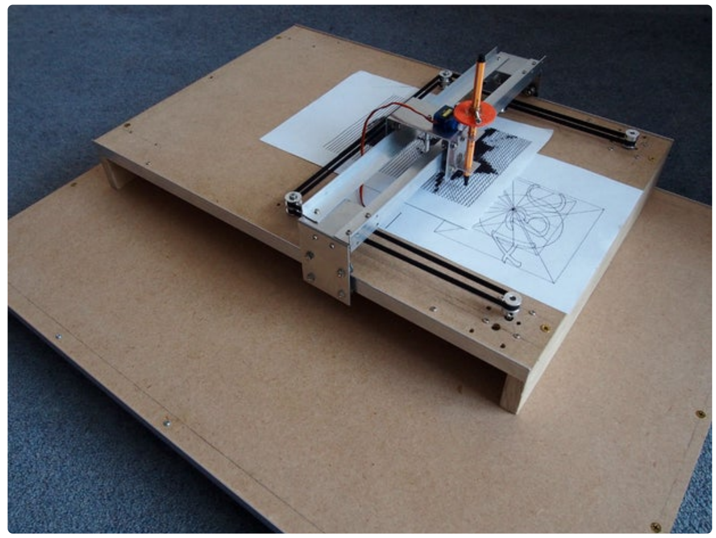
[2]

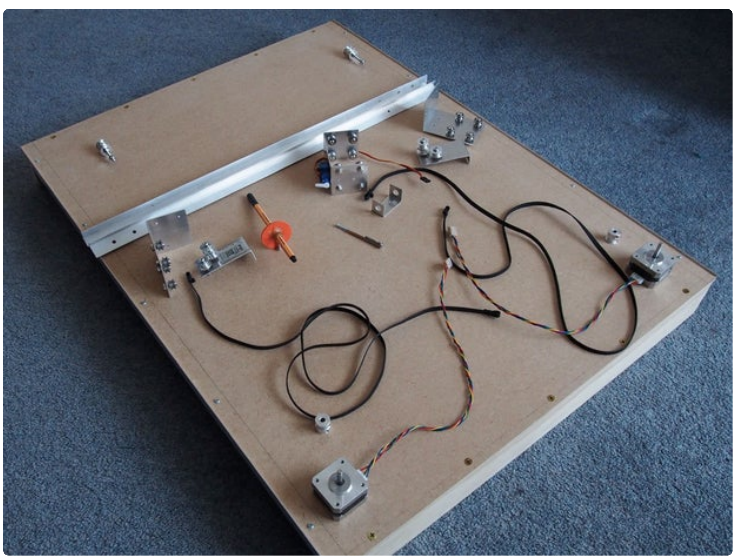
CoreXY requires two full-length timing-belts. Halving the original H-bot timing belt allowed the coreXY concept to be tested before longer timing belts were ordered. It also accounts for the reduced plot area shown in photo 2.


Motor A
Motor B


Motor A
Motor B


Motor A





Step 7: Construction ... the Base

Decide on your plot area:

- An A4 plotter requires a 6mm x 600mm x 400mm piece of particle board.
- An A3 plotter requires a 6mm x 800mm x 600mm piece of particle board.
- or larger ... the design is scalable

Bolt a length of 1.5mm x 20mm x 20mm aluminium "Tee Section" extrusion along opposite sides of the base.
Ensure that both rails are parallel.

The extrusions should be flush with the long edges of the base and positioned under the base such that the edges just protrude (see photo). These extrusions act as railway lines for the overhead gantry.

Screw 65mm x 18mm timber end supports to the base. The timber supports will need a rebate for the aluminium extrusions ... make a shallow cut then knock a sliver of timber out using a chisel positioned against the end grain.

Step 8: Adding the Fixed Pulleys and Motors

The two timing-belts are stacked vertically above each other.

This is achieved as follows:

- One motor pulley is inverted as shown in photos 1, 2, & 3.
- The toothed idler pulleys are stacked in sets of two as shown in photos 4 & 5.

Mounting the motors:

Drill four x 3mm mounting holes, and one spindle clearance hole, for each NEMA17 stepping motor as shown in photos 2 & 3.

The outside 3mm holes are 25mm from each edge of the base [1]. This distance ensures the the motors avoid the wooden support and the aluminium side rails.

Mounting the fixed pulleys:

Drill two 4mm holes for the fixed pulleys.

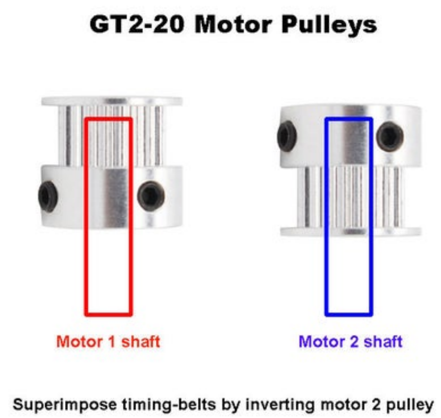
These holes are located 25mm from the opposite end to the motors and 40.5mm from the side rails. This distance ensures that the pulleys avoid the wooden support. It also ensures that the timing-belts are parallel with the side rails.

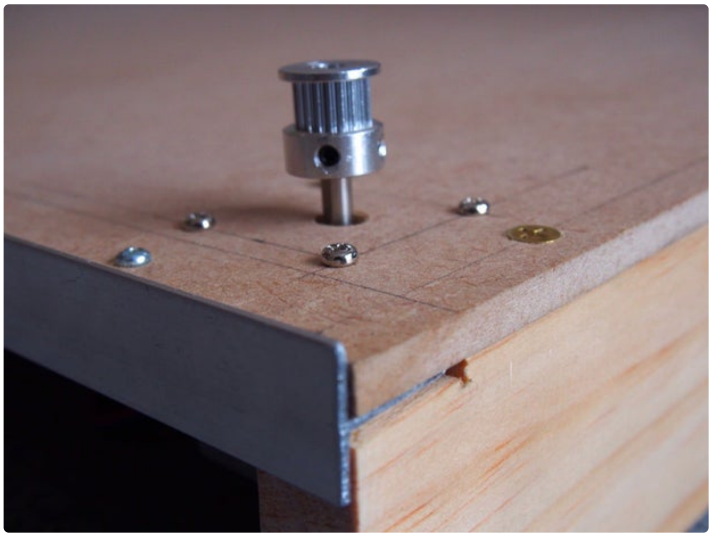
Key points:

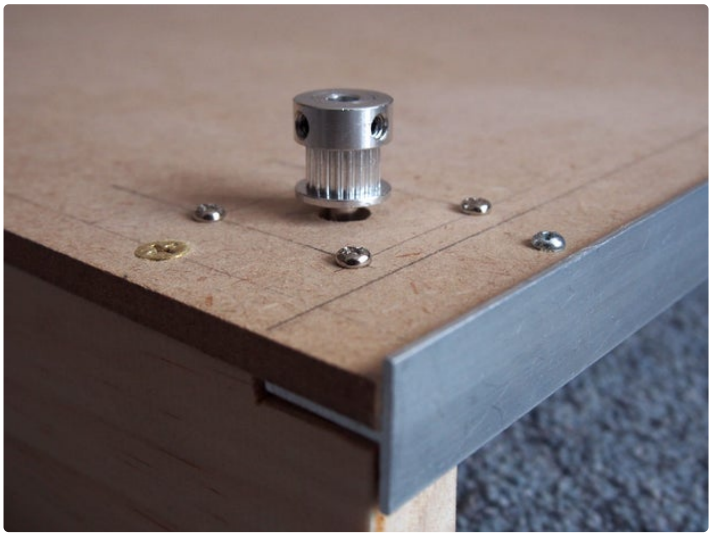
- Timing belts stretched between each motor and the associated fixed pulleys MUST be parallel to the side rails.
- Separate the pulleys with a 4mm washer. The washer prevents the pulley edges from rubbing.

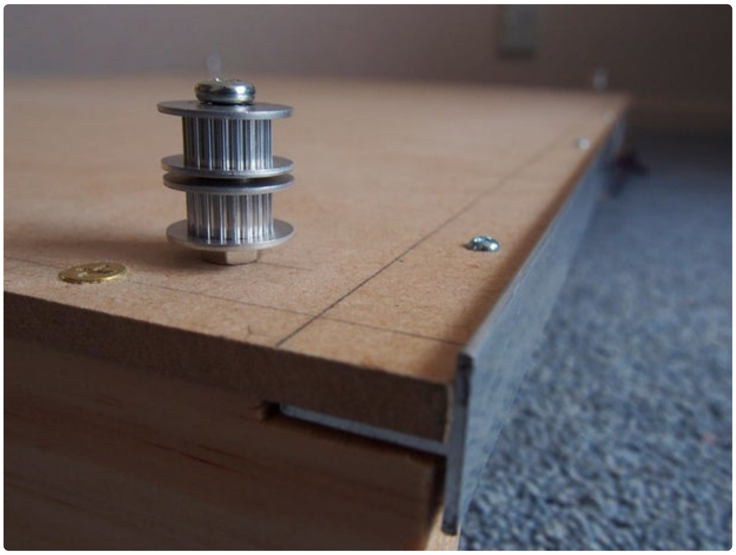
[1]

The 3mm motor mounting holes are spaced 31mm apart. This means that each spindle is 40.5mm from the nearest siderail.


GT2-20 Motor Pulleys
Motor 1 shaft
Motor 2 shaft
Superimpose timing-belts by inverting motor 2 pulley







Step 9: The Gantry

Vertical (Y-axis) motion is provided by a movable gantry that runs along the fixed side-rails attached to the base.

Horizontal (X-axis) motion is obtained by running a pen carriage-assembly across two rails attached to the gantry end brackets.

The gantry wheels are fashioned from VZ624ZZ V-groove sewing machine pulleys (photo 3). The V-groove prevents the gantry moving sideways.

End brackets:

Cut two 60mm x 130mm brackets from a sheet of 18 gauge aluminium using the method described in instructable https://www.instructables.com/id/How-to-Cut-Fold-S...

Drill two 3mm holes and four 4mm holes at the locations shown in photo 1.

Fold a 50mm top for each bracket.

Mounting the wheels:

Sandwich four "V-groove" pulleys between a 4mm nut and bolt. The nut prevents the sides of the pulleys touching the aluminium.

Now bolt these wheels through the 4mm diameter holes as shown in photo 2.

Attaching the end brackets to the base

Attach a gantry end plate to each side of the plotter.

To do this:

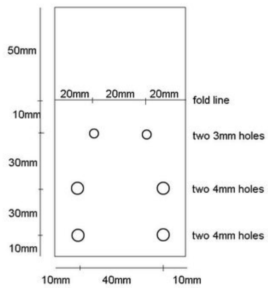
- the grooved wheels must straddle the side-rails.
- elongate the bottom holes as required to eliminate any vertical play in the gantry.

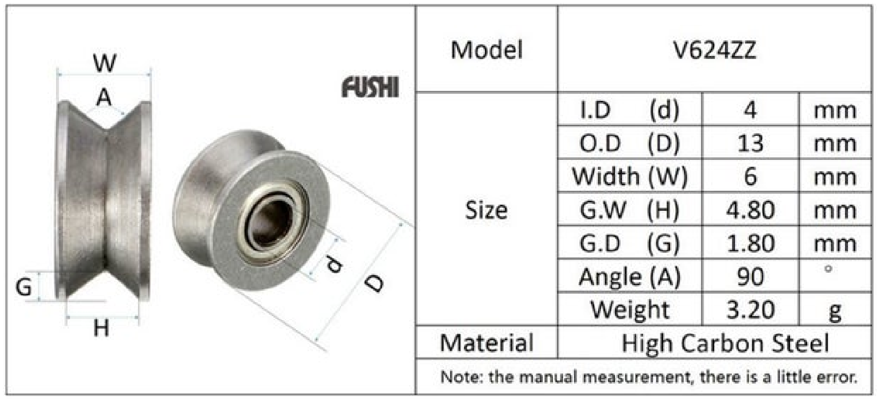
When correctly adjusted the gantry end-brackets should roll freely when the base is tilted slightly.

Top rails:

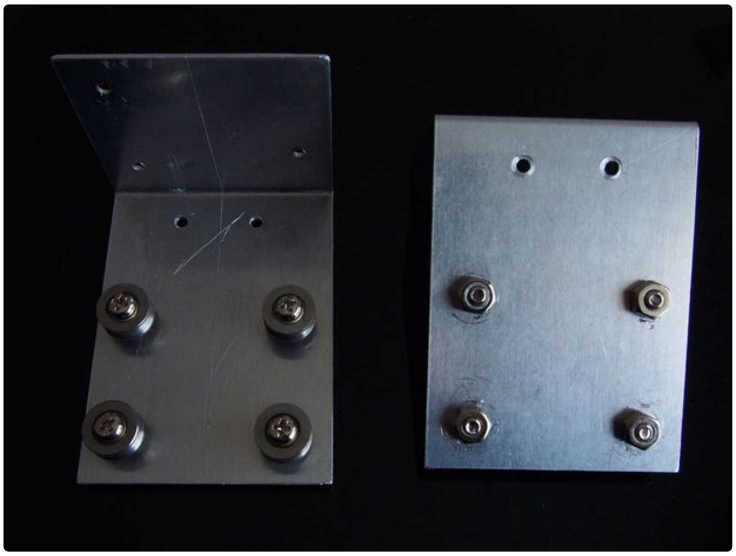
The over-head gantry rails are attached when we attach the pen carriage-assembly ...

Gantry Brackets





Model	V624ZZ		
Size	I.D (d)	4	mm
	O.D (D)	13	mm
	Width (W)	6	mm
	G.W (H)	4.80	mm
	G.D (G)	1.80	mm
	Angle (A)	90	°
	Weight	3.20	g
Material	High Carbon Steel		
Note: the manual measurement, there is a little error.			



Step 10: The Gantry Pulley Brackets

The gantry pulley brackets:

The gantry pulley brackets are cut from 18 gauge aluminium sheet. An approximate [1] drilling template is shown in photo 1.

Photo 2 shows the pulleys attached to the motor 1 (left-hand) bracket.

Photo 3 shows the pulleys attached to the motor 2 (right-hand) bracket.

Key points:

- The smooth side of the timing belts always sees a smooth pulley.
- The ribbed side of the timing belts always sees a toothed pulley.
- One of the double pulleys on each bracket is NOT used ... it simply acts as a spacer.

Attaching the pulley brackets to the gantry:

Press each pulley bracket against the inside of the gantry bracket such that the bracket is centered and the heads of the pulley bolts are clear of the base.

Mark these positions by means of a pencil through the two existing holes in each of the gantry brackets.

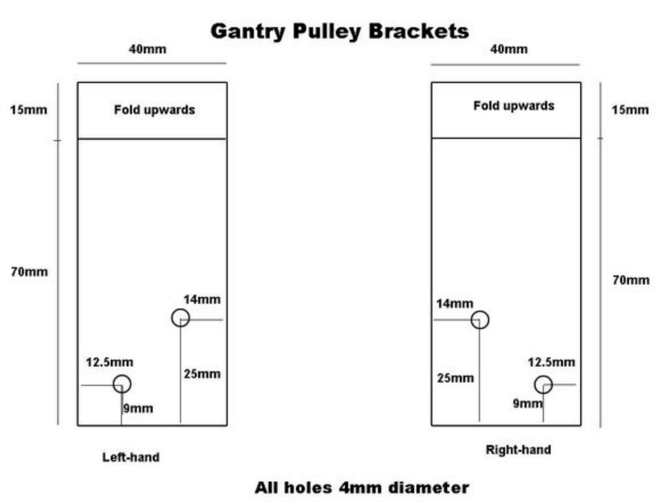
Now drill 3mm mounting holes and bolt the bracket pairs together.

[1]

The 4mm hole positions for the idler wheels depend on the sharpness of the upturned folds.

Fold each pulley bracket then custom position the holes such that:

- both timing-belts over-lap when viewed from above.
- the idler wheels used as spacers don't touch any belts.

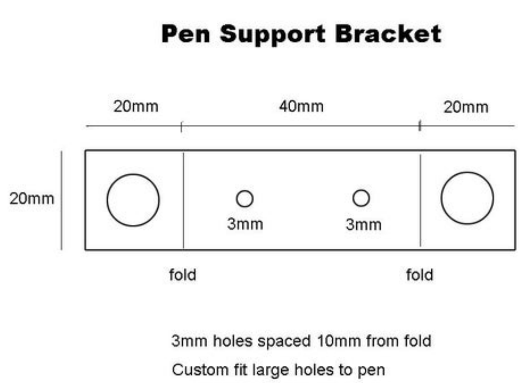


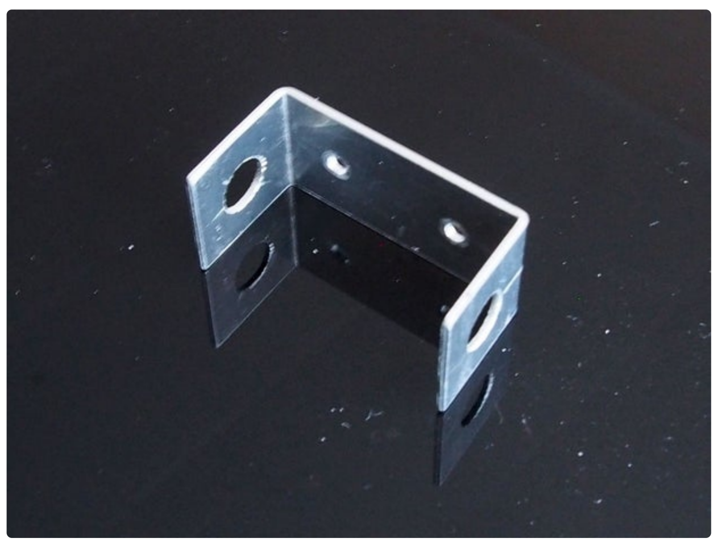
Step 11: Pen Support

The pen support is made from 18 gauge aluminium sheet. The dimensions are shown in photo 1

Custom size the large holes such that your pen slides freely but without any sideways wobble.

Attach a pen-lift collar to your pen. Mine was made from the brass fitting found inside a radio knob and a circular piece of plastic but anything will do ... so long as the position can be adjusted.





Step 12: Pen Carriage Assembly

The pen carriage is made from 18 gauge aluminium sheet. A drilling template (not to scale) is shown in photo 1.

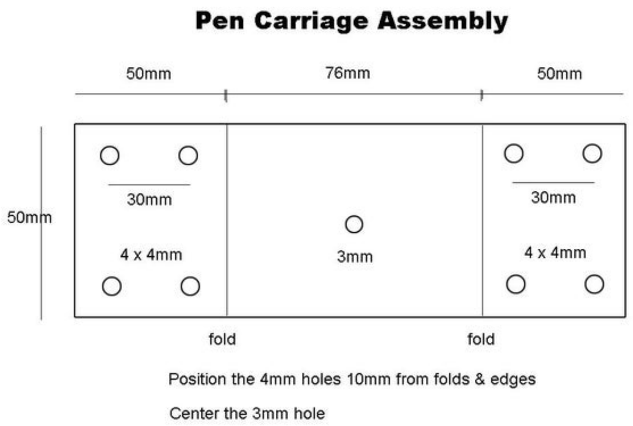
Assemble as follows:

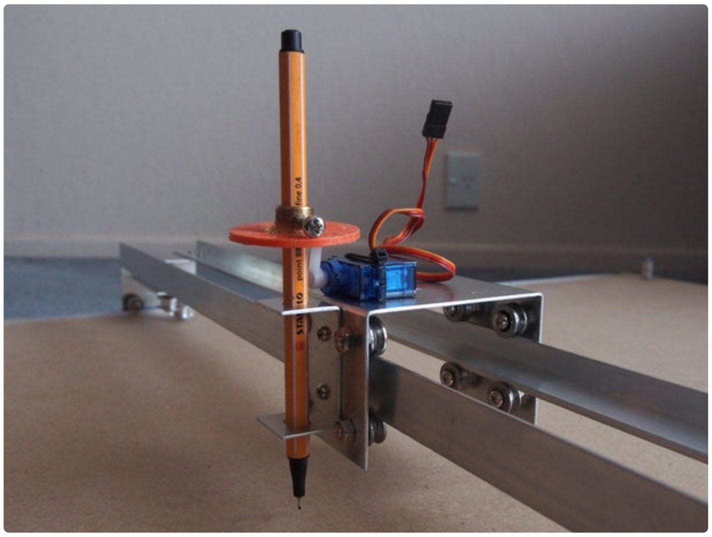
- drill and fold the metal
- attach 8 x v-groove pulleys as shown in photo 2. Elongate the lower holes such that there is no vertical play when the aluminium "tee section" rails are fitted.
- use a cable-tie to attach the SG90 servo as shown in photo 2. Drill the cable-tie holes to suit.
- use the 3mm holes in the pen-support as a drilling template when attaching the pen.

Fitting the top gantry rails:

Attaching the rails:

- Pass two 600mm lengths of aluminium "tee section" between the "V-groove" pulleys as shown in photo 2.
- Attach each gantry bracket to one of the rails by means of two 3mm nuts and bolts.
- Position the second rail such that the pen carriage-assembly moves freely. Now drill and bolt the rail to the gantry brackets.





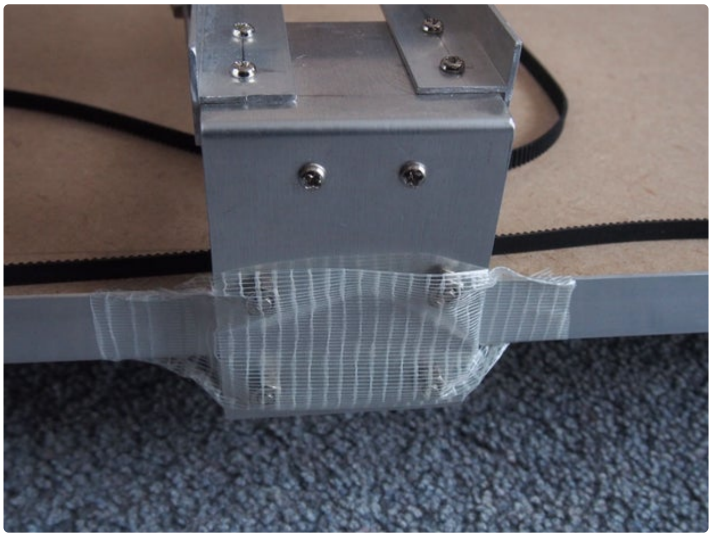
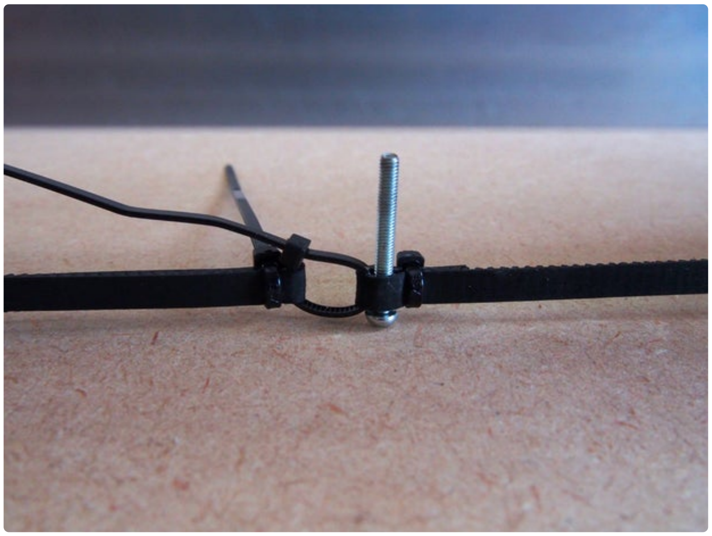
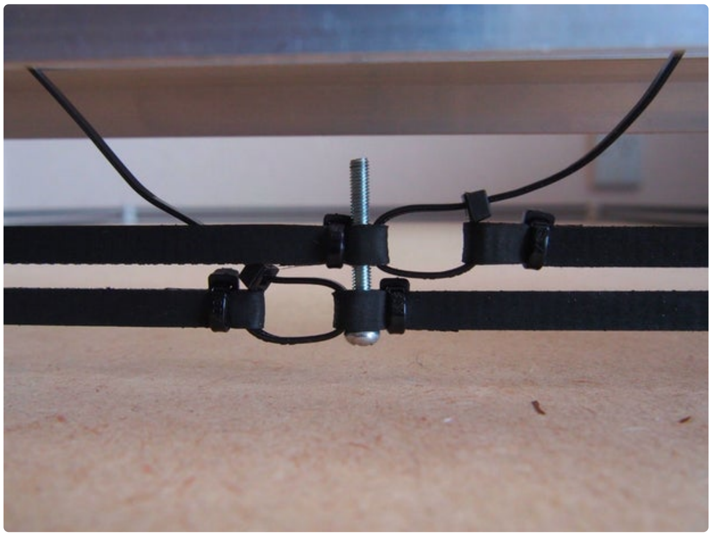
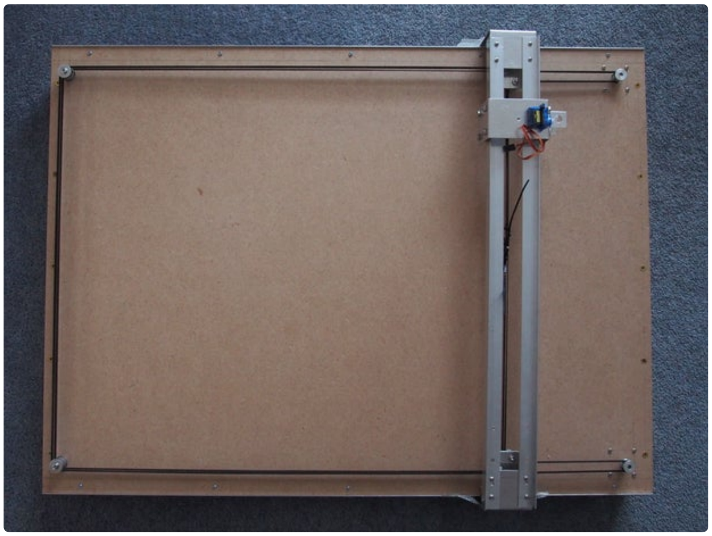
Step 13: Attaching the Timing Belts

Key points:

- the plotter has TWO timing belts.
- each timing belt forms a continuous loop (see photo 2)
- the timing belts are mounted at different heights ... the pulleys share a common shaft.
- the timing belts are tensioned using cable-ties.
- both timing belts are attached to a common 3mm bolt (see photo 3)

Method:

- push the pen carriage-assembly to one side.
- tape both gantry ends to prevent movement (photo 1).
- thread the bottom timing belts and fix the ends (photo 2).
- thread the top timing belt and fix the ends (photo 3).
- the timing belts should look like this (photo 4)
- remove the tape.
- tension the belts such that the carriage-rails are the parallel to the base end.

Step 14: Attaching the Pen Carriage Assembly

Two 25mm long threaded spacers are joined together (by means of a 3mm head-less bolt) to form a solid rod.

The belts are attached to this rod as shown in photo 1.

Attach the top of the rod to the center of the pen carriage-assembly using a 3mm bolt.

Testing:

The pen should:

- move to the left when BOTH motors are turned clock-wise.
- move to the right when BOTH motors are turned counter-clockwise.
- move diagonally if only one motor is rotated.

Step 15: The Menu

Upload coreXY_plotter.ino to your arduino if you haven't already done so. Instructions for this are given in step 2.

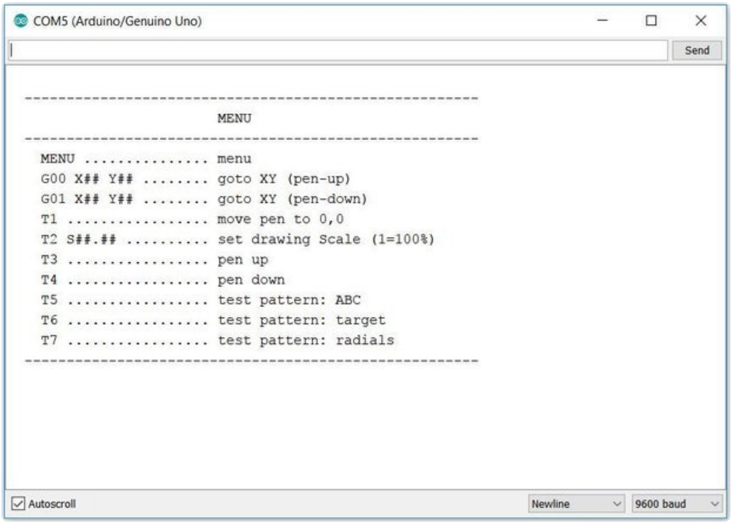
Connect a USB cable to your arduino and left-click "Tools|Serial Monitor" ... a menu similar to photo 1 should appear.

Apply 12 volts power to your motors and your plotter is ready to use.

The menu is not case sensitive. Typing:

- MENU brings up the menu
- G00 allows you to send the pen to a specific XY co-ordinate with the pen raised.
- G01 allows you to send the pen to a specific XY co-ordinate with the pen lowered.
- T1 allows you to position your pen over your 0,0 co-ordinate. Type 'E' to exit.
- T2 allows you to scale your drawiing. For example "T2 S2.5" will scale your drawing 250%. The default scale is 100%.All pen moves use the drawing scale last set using this menu option
- T3 and T4 allow you to raise or lower the pen.
- T5 draws an "ABC" test pattern.
- T6 draws a "target".
- T7 draws a set of radial lines

The internal wiring of some motor brands are reversed. If your motor directions are reversed then use the alternate code in the step_motors() function.



Step 16: Creating and Sending Gcode Files to Your Plotter

This plotter assumes that co-ordinate (0,0) is at the lower-left corner of your paper. As such it is 100% Inkscape compatible.

Instructions for creating gcode files are given in the following instructables:

- https://www.instructables.com/id/CNC-Robot-Plotter... (step 9)
- https://www.instructables.com/id/CNC-Edge-Detectio...
- https://www.instructables.com/id/Make-Your-Own-CNC...

Instructions for sending your gcode file to this plotter are given in instructable:

- https://www.instructables.com/id/CNC-Gcode-Sender/

Click here to view my other instructables.

Up-scaled version of Lingib's COREXY plotter

https://www.youtube.com/watch?v=YiVmjRIDAq8

Thank you for your message :)
Congratulations on up-scaling your plotter ... it's working well :)

I have purchaced big easy drive its look different, please help how to wiring?

The reason the module looks different is that your module is an "Easy Driver" not a "Big Easy driver".

Both modules are similar but the "Big Easy Driver defaults to 16 x microstepping whereas the "Easy Driver" defaults to 8 x microstepping.

If you wish to use the "Easy Driver" you will meed to change line 43 in the *.ino code to read:
#define STEPS_PER_MM 200*8/40 //200steps/rev; 8 x microstepping; 40mm/rev

Apart from this minor code change the wiring, and motor current adjustment(s), is the same as that set out in Step 1 of the instructable.

Each of the pinouts for the "Easy Driver" are axplained in
https://media.digikey.com/pdf/Data%20Sheets/Spark...

Each of the pinouts for the "Big Easy Driver" are explained in
http://www.schmalzhaus.com/BigEasyDriver/BigEasyD...

Ok Thanks, but where i have to connect two wire coming from SG90?

Hi, I have done everything up to send G_code. I have type the file name in command panel and it not started moving stepper.

Things to try:
(1)
Confirm that your plotter works as follows:
- Launch your Arduino IDE,
- click "Tools|Serial Monitor",
- then issue some manual commands.
- close your Arduino IDE
(2)
Now send a known good file to your plotter
- download and copy Letter_B.ngc into your C:\...\processing3_terminal folder
- launch "processing3_terminal.pde"
- now follow the instructions in Step 3 of the "CNC Gcode Sender" instructable
(3)
Now try your aa.ngc file
- copy your aa.ngc file into your C:\...\processing3_terminal folder
- mimic the the instructions in Step 3 of the "CNC Gcode Sender" instructable using aa.ngc instead of Letter_B.ngc
If steps (1) and (2) work and (3) doesn't then the problem is with your aa.ngc file.

The SG90 has three wires.
Just follow the wiring shown in Photo 1,Step 3.
The orange wire goes to pin D3 of the Arduino.

The red wire goes to 5V on either the Easy Driver or the Big Easy Driver.
The brown wire goes to GND on either the Easy Driver or the Big Easy Driver.

Thanks Lingib!

Iam unable to load sketch, It showing bellow error
--Arduino: 1.6.1 (Windows 7), Board: "Arduino Uno"
coreXY_plotter.ino: In function 'void setup()':
coreXY_plotter.ino:104:6: error: redefinition of 'void setup()'
sketch_feb14b.ino:1:6: error: 'void setup()' previously defined here
coreXY_plotter.ino: In function 'void loop()':
coreXY_plotter.ino:174:6: error: redefinition of 'void loop()'
sketch_feb14b.ino:6:6: error: 'void loop()' previously defined here
Error compiling.

This report would have more information with
"Show verbose output during compilation"
enabled in File > Preferences.

I am unable to replicate your problem (screen shots attached)

I notice that you appear to be using Arduino version 1.6.1 and Windows 7. I suggest that you update your software to the latest Arduino version for your operating system. The current Arduino version for Windows 10 is version 1.8.8.

There are two versions of my code in Step 2 of this instructable. Try each of these files. The reason for two files is explained in my comment dated 22 December 2018.

You have possibly copied the *.ino code into your "sketch_feb14b" twice as your error messages read: "coreXY_plotter.ino:104:6: error: redefinition of 'void setup()'"

The following method should eliminate any copying errors:
(1) download the file "coreXY_plotter_1a.ino" into your download folder
(2) double-left-click "coreXY_plotter_1a.ino" using Windows File Explorer.
(3) this should launch the Arduino IDE which will then ask if it can place the file into a folder of the same name.
(4) click "OK"
(5) now upload the file to your Aduino ... all going well it should compile.


Hi
CPS-3205 power supply not avilable in my place, So alternativly can i use 12V 3A power adafter? or Please suggest me.
Also 12v power suply need to connect only Big easy driver? please give melittle more details.

Any 12 volt DC power adapter capable of 1 amp should be fine providing that you adjust each motor current to 400 milliamps (0.4 amps) as described in Step 1, "Adjusting the motor current(s)".

Since power adapters do not have an inbuilt current meter you will need to (temporarily) insert an amp-meter in series with the power supply while you adjust each of the Big Easy Driver current limits to 0.4 amps (400 milliamps). Before adjusting the Big Easy Driver currents use an ohm-meter to check that each of the two motor coil windings are adjacent to each other on the Big Easy Driver boards.

The 12 volt supply is ONLY connected to the Big Easy Drivers as shown in the CoreXY Plotter wiring diagram. The 5 volts for the Arduino Uno R3 is obtained from the USB cable connected to your PC.

Thanks for reply, I got now. 0.4amps should out from each big easy drive and 0.6amps out for SG90.

Amazing project, very nice and clean.

I have one simple question, in the draw_line() function, is the plane divided into octants or quadrants.

Many thanks for the project.

My draw_line() function is a variation of Bresenham's Line Drawing Algorithm and uses "octants" as defined in the attached diagram.

Bresenham's algorithm is only valid for "octant 0" which means that you have to convert each XY coordinate to "octant 0" for processing, then convert the results back to the original octant.

There are also numerous software solutions, one of which is derived in my instructable https://www.instructables.com/id/CNC-Drum-Plotter/ (Step 5)

In that instructable I used switch() functions, each with eight conditions, to encode and decode each of the eight octants. This works but is relatively slow as each plot requires up to twenty-four comparisons (eight for encoding; eight for decoding the X axis; and eight for decoding the Y axis).

The algorithm that I have developed for this plotter recognizes that a circle not only contains eight "octants" but also has four "quadrants":

- quadrant 0 contains octants 0,1

- quadrant 1 contains octants 2,3

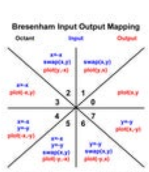
- quadrant 2 contains octants 4,5

- quadrant 3 contains octants 6,7

If you study the attached diagram you will note that dX is always positive in "quadrants" 0,3 and that dY is always positive in "quadrants" 0,1. Now add a "swap" flag and the number of comparisons needed to plot each point in the draw_line() function reduces to eight which is significantly faster.

Hopefully this clarifies the following comment which appears in my draw_line() function:

"The algorithm automatically maps all "octants" to "octant 0" and automatically swaps the XY coordinates if dY is greater than dX. A swap flag determines which motor moves for any combination X,Y inputs. The swap algorithm is further optimised by realising that dY is always positive in "quadrants" 0,1 and that dX is always positive in "quadrants" 0,3."




Yes sir, I will purchase Big easy driver right now. Thank you

Hi Lingib,

Thank you very much for this instructable and all the support you've been giving to people since then. Not only is admirable to create amazing projects but also to be able to support them afterwards;

Together with a friend we've been trying to build a scaled version of this project (to print around 70x100cm);

As we are also trying to build on top of the software you've created for this project, we've been facing a problem that we couldn't solve so far. When we tell the pen to move to big coordinates (X500 Y700 - 50cm,70cm -, for example) the pen inverts its own movement, going to -X500, -Y700 instead. We printed the values that we are sending to the motors and those values are correct (positive) but it seems that there is something that is telling the motors to go the opposite way. Did you implement any kind of flag to trigger this or do you know how to solve this issue?

Thanks again for all the support!

Thank you for your interest in my CoreXY plotter :)

The fact that correct numbers are being sent to the motors indicates that your motor directions are reversed ... I have allowed for this in my code as some motors (depending on brand) have their internal wiring reversed.

To reverse your motor directions go to the step_motor(){...} subroutine and change this code segment

// ----- set motor directions

//(DIRECTION1 == CW) ? SET(pattern, dir1) : CLR(pattern, dir1); //normal motor direction

//(DIRECTION2 == CW) ? SET(pattern, dir2) : CLR(pattern, dir2); //normal motor direction

(DIRECTION1 == CCW) ? SET(pattern, dir1) : CLR(pattern, dir1); //motor windings reversed

(DIRECTION2 == CCW) ? SET(pattern, dir2) : CLR(pattern, dir2); //motor windings reversed

to read

// ----- set motor directions

(DIRECTION1 == CW) ? SET(pattern, dir1) : CLR(pattern, dir1); //normal motor direction

(DIRECTION2 == CW) ? SET(pattern, dir2) : CLR(pattern, dir2); //normal motor direction

//(DIRECTION1 == CCW) ? SET(pattern, dir1) : CLR(pattern, dir1); //motor windings reversed

//(DIRECTION2 == CCW) ? SET(pattern, dir2) : CLR(pattern, dir2); //motor windings reversed

Hopefully this solves your problem.

Hi, I have a problem with the stepper motors, I got two 17hs3430 from AliExpress exactly as suggested, and two Big EasyDriver. When I try the program, the steppers seem to move randomly, especially when I try to move them using the G00 command they don't move correctly or at all. I'm not sure if the cause of this behavior is because the motors are broken or because or something else. They seem to have very little torque when they (almost randomly) move. The rest of the machine seems to be constructed correctly: if I move the motor manually the movement of the pen is correct, but the motors are so weak that they can't move the pen at all. Any suggestion is more then welcome.

Each motor has two windings. Identify the leads that are attached to each winding (use an ohmeter on resistance) and connect the two leads side-by-side as shown on the EasyDriver screen print..

Swap the two center wires if the coil wires are not adjacent. Hopefully this will solve the problem.

That worked! My CNC Plotter works perfectly now, it's a really awesome project, thanks again!

Your plotter sounds great ... glad to have been able to help :)

You may also find these instructables to be of interest as they build upon your CoreXY design:
https://www.instructables.com/id/CNC-Pen-Lift/
https://www.instructables.com/id/Make-Your-Own-CNC-Plotter-Image/
https://www.instructables.com/id/CNC-Pen-and-Wash-Portrait/

Hey lingib,
Thanks so much for your instructable, I have found it very useful. Thought I would share with you how my attempt at building a remix of your design has gone. So far the results are pretty good, only stumbling block at the moment is circles. Any ideas here, I think it might be something to do with belt tension. The one major difference is that I haven't used your code but have instead used grbl, kudos to you though for writing your own code, that's incredible!

Anyway, thanks again for your instructable!

Glad that you found my instructable useful. Thank you for sharing ... you've just made my day :)

Fine on porting GRBL to this design.

Can you be a bit more specific about the circles. Are they round, elliptical, or don't the ends meet? How do GRBL circles compare with the same circles using my code?

The following quote from https://github.com/grbl/grbl/wiki may be a clue.

"CoreXY Support: Grbl now supports CoreXY kinematics on an introductory-level. Most functions have been verified to work, but there may be bugs here or there. Please report any problems you find!"

I'm pretty sure the problem was belt tension. I strengthened the X carriage by adding a bottom cross support which made it more difficult to attach all the belts, I think this was a catch 22 situation.

I must admit I have now abandoned the CoreXY setup and gone with a more conventional setup where there is a motor mounted on the gantry to move the X carriage. I am now getting better results when drawing circles.

I never actually got your code working on my setup, I have different drivers but as you have stated it should work with easydrivers. However I wasn't very persistent with my attempt.

I am really glad to have been introduced to the CoreXY system though, it really is cool how you avoid mounting steppers on the gantry.

And thank you for sharing your results :)

Your video is great. Am impressed with your mechanical setup, choice of rails, motor mounts, and your pen mechanism ... well done!

Here's a youtube clip of another video so downloading isn't necessary:
https://www.youtube.com/watch?v=UCYlUPE-esU

Fantastic :)

I like the way you have mounted the motors.

Hey lingib!
Thanks for the effort of making this instructables, the project is great!

We built the plotter and is working well, both the mechanical and software side.
We are interested to add a *bounds function* that allow us to go back to the same origin every time we turn on the plotter.

Have you ever thought about this?
Do you have any suggestion of where to start?

Thanks!

lingib (author) capitanlerryReplya few seconds ago

Thank you for your comments ... congratulations on your project ... it sounds great :)

In the interests of simplicity I left out a number of "nice-to-have" features;

"Page-boundaries" should be easy to implement in software. Stop the motors when the pen strays outside the page area but let the calculations for X and Y continue.

"Auto-return-to-a-specified-coordinate-on-startup" is a bit more complicated as the software doesn't know where the pen is positioned when the power is applied ... a reference point needs to be created. Adding a limit-switch to both the "gantry" and the "pen-carriage is one possible solution.

On power-up make the software move the pen left until the carriage limit-switch is activated. Flag this point as X=0 and temporarily prevent any further motion to the left.

Now move the pen downwards until the gantry limit-switch is activated. Flag this point as Y=0 and prevent any further down-ward motion ... we now have our physical reference.

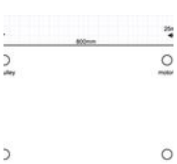
Offsets may now be added to these XY values if you want to position your paper in the middle of the plotter.

Extra limit-switches for to prevent the plotter trying to move too far upwards and to the right would also be a good idea.

I'm having some issues understanding the distance of the pulleys/motors from the borders, you wrote:

Drill two 4mm holes for the fixed pulleys. These holes are located 25mm from the opposite end to the motors and 40.5mm from the side rails.

but the distance of the motors from the border are 25mm each side, so that would mean that the motors and the pulley have different distances from the side rails. I'm attaching a little drawing to try yo understand what I think is correct, but I still don't get one measure (see the ? in the picture). Can you tell me what I'm missing?



> "Mounting the motors

> Drill four x 3mm mounting holes, and one spindle clearance hole, for each NEMA17 stepping motor as shown in photos 2 & 3.

> The outside 3mm holes are 25mm from each edge of the base. This distance ensures the the motors avoid the wooden support and the aluminium side rails."

The 3mm motor mounting holes are spaced 31mm apart. This means that each spindle is 40.5mm from the nearest siderail

> "Mounting the fixed pulleys

> Drill two 4mm holes for the fixed pulleys.

> These holes are located 25mm from the opposite end to the motors and 40.5mm from the side rails. This distance ensures that the pulleys avoid the wooden support. It also ensures that the timing-belts are parallel with the side rails."

The fixed pulleys must be 40.5mm from the nearest siderail if the timing belts are to be parallel.

The distances from the wooden end supports is arbitary ... so long as the motors and nuts don't touch the wood.

Hope this helps :)


It does, thanks!

You're welcome ... good luck with your project :)


Thanks! I'll keep you posted, I'm pretty sure I'll have some other questions in the future ;)

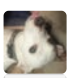

Really impressive instructable, thankyou! Could anyone give me some pointers on sourcing parts? Having trouble finding some of them anywhere (I'm in Canada). Including but not limited to the aluminum tee's.

Thank you for your comment :)

Download the file CoreXY_parts_list.txt attached to Step 3. This file lists the website from which all items were obtained and the product descriptions.

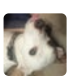
Cut and paste each of the descriptions into the website's search-bar and select the supplier who offers the best deal.

Two thin aluminium 'L-sections' bolted back-to-back should work If you can't find aluminium 'tee-sections'. Most hardware stores contain a range of aluminium extrusions.


No video attached!

The instructable definitely contains a video.

Try closing and reopening your browser as weird things can happen.


Is it me or is this plotter able to plot a 45 degree angle line or any lines not just horizontal or vertical? How are the motors to turn to plot the angle lines?

CoreXY motion occurs due to differential action:

- rotating motor 1 counter-clockwise by hand will cause the plotter to move UPWARDS to the right at an angle of 45 degrees.

- rotating motor 2 counter-clockwise by hand will cause the plotter to move DOWNWARDS to the right at an angle of 45 degrees.

- when both motors rotate counter-clockwise the up and down movements CANCEL and the plotter moves horizontally to the right.

When plotting angle lines one of the motors runs slower than the other allowing for angles other than 45 degrees.

The video attached to this instructable shows the plotter in operation.

hi, please help I’m having trouble running the stepper motors ...whatever command I sent through universal Goode sender both the x and y axis steppers rotate , when I press y axis both of them rotate in same direction and when x axis is pressed both rotate in opposite direction.... I’ve been trying to sort out this issue, I tried using rabbit code sender, changed a4988 stepper drivers, cleared eeprom , flashed Grbl multiple times , tried changing the comport,tried different baud rate, nothing seems to work ..I’m stuck with this problem and when I stream a Gcode file both motors struggle due to multiple commands like when x axis code is running y axis also rotates ,at the same time if Yaxis command is given motor struggles making grinding noise... please provide a solution . What could have possibly went wrong ,should I change Arduino or CNCinfusion shield v3 ?

My setup:

Arduino Uno

Nema 17 4.2kg/cm

A4988 stepper driver

Arduino CNC shied v3

Tower pro mini servo

Power supply: 19v 4.7 A

Grbl 0.9i

Is your plotter a CoreXY Plotter ... I ask this because you mention items not in this article such as GRBL and Universal Gcode Sender?

The good news ... your motors appear to be working correctly.

Horizontal movement occurs when BOTH motors rotate in the same direction. Vertical movement occurs when BOTH motors rotate in opposite directions.

You may try loosening the belts ... the motors can stall and produce a growling noise if the belts are over-tightened.

Try talking directly to your arduino using the serial monitor in your arduino IDE. Only move to GRBL and Gcode sender when you have each function on the test menu working.

[delete]

you know what ..regardless of what command I enter whether it’s x or y both x and y rotate that’s the problem

CoreXY plotters differ from conventional plotters in that BOTH motors rotate when moving along the X and Y axis. An excellent explanation may be found at http://corexy.com/theory.html. (See instructable steps 4, 5)

CoreXY motion occurs due to differential action. Try this:

- rotate motor 1 counter-clockwise by hand ... the plotter should move to the right and up.

- rotate motor 2 counter-clockwise by hand ... the plotter should move to the right and down.

- now rotate both motors counter-clockwise by hand ... the plotter should only move to the right.

If this works then your plotter belts are correctly threaded.

I get the impression that you may have substituted my code for GRBL in which case you will have to configure GRBL 0.9i (or later) for coreXY. I have not tried this so am unable to help.

You also appear to have substituted the stepper drivers. Providing your substitute board(s) are Big EasyDriver compatible this shouldn't be a problem.

On re-reading your question I see some potential issues:

1 - my plotter does not use a motor shield.

2 - my software is self-contained ... 3rd party libraries are not required.

It would appear that some of the pins to your motor shield are incorrectly mapped or your motor connections are swapped as HORIZONTAL motion requires both motors to turn in the same direction and VERTICAL motion requires that both motors rotate in opposite directions.

The logic I have used when stepping the motors assumes that each motor controller has a "direction" pin and a "step" pin. The logic level for each direction pin is set then both motors are SIMULTANEOUSLY stepped.

You may have to remap the "direction" and step "pins" to match your motor shield. These pins are defined in the arduino header.

Another question: is 16 gauge aluminium sheet going to be enough? I'm having a hard time finding the 18 gauge.

16 gauge aluminium should be fine.

I only used 18 gauge aluminium because it was available.

Hi! Awesome project! I have a question for you: what is the bore I need to select for the 2GT Idler Pulley? I can select 3, 4 or 5 mm, and I can't find the correct bore size in the description. Thanks!