

Project Convert to Paint

In no particular order, this document describes parts of our project that you may find useful for your own robotic art project.

Universal artist tool chucks:

The universal artist tool chucks that we have developed during this project can hold pens, pencils, paint brushes, and crayons of almost any size. The tool chucks use three adjustment screws to align the tool in the chuck.

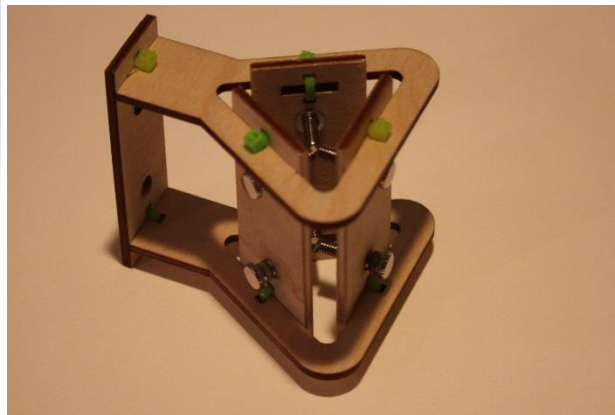
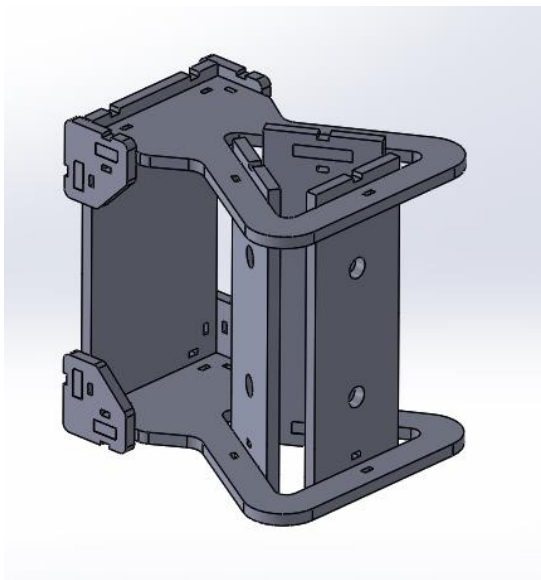
Tool chuck without swivel:

The universal artist tool chuck without swivel will cost \$40-\$50.

From the local hardware store you will need a package of 4in cable ties and six $\frac{3}{4}$ " 8-32 thumb screws.

From McMaster-Carr you will need a pack of 8-32 threaded inserts from (part # 94738A110).

You will need to determine the best way to secure the tool chuck to your cnc machine. I have left the mounting plate without any holes so it will be easier to drill whatever hole pattern that works best for your machine.



Tool chuck with fourth axis swivel:

The universal artist tool chuck with swivel will cost \$75-\$95.

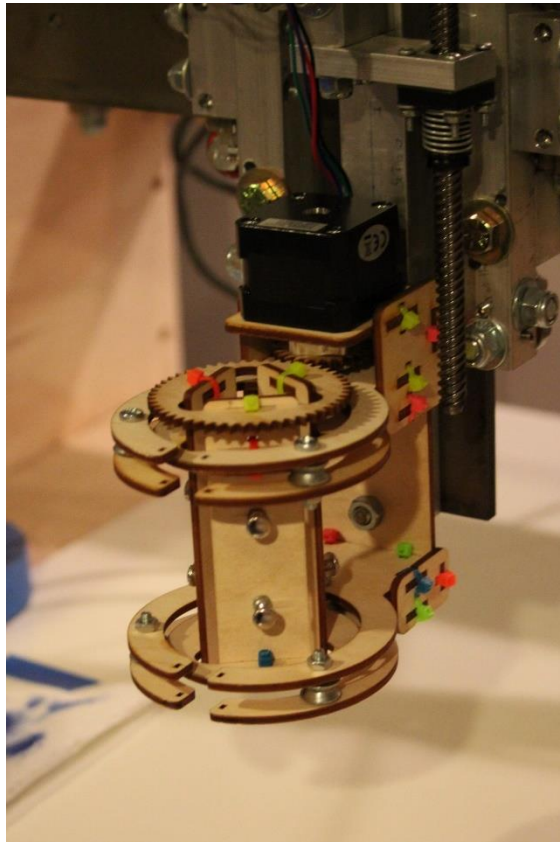
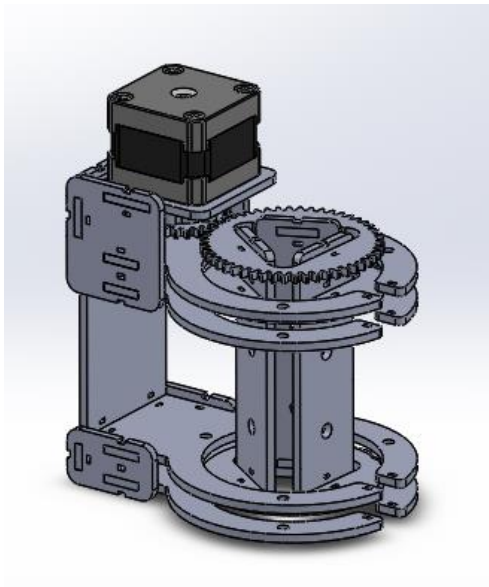
From the local hardware store you will need a package of 4in cable ties, six $\frac{3}{4}$ " long 8-32 thumb screws, six 20mm long 4mm machine screws, six 4mm nuts, and twelve 4mm lock washers.

From McMaster-Carr you will need a pack of 8-32 threaded inserts (part # 94738A110).

From Amazon.com you will need six 6mm x 13mm v bearings, a 5mm bore set screw hub, and a NEMA 17 stepper motor. The stepper motor could be as low as 64 oz-in; it doesn't need much torque.

From Amazon.com you will also need a stepper motor driver. If you are not familiar with what these are just google it. Others will have much better explanations.

You will need to determine the best way to secure the tool chuck to your cnc machine. I have left the mounting plate without any holes so it will be easier to drill whatever hole pattern that works best for your machine.



The Convert to Paint Process - Vector Image to Painting Gcode:

This section will detail the process for converting a bitmap photograph to painting g-code. Using the "Project Convert to Paint" process, any vector image can be converted to a painting. That is not to say that any vector image will produce a good painting. By walking through our process step by step we hope the reader will get a better understanding of what it takes to "Convert to Paint."

Couple things to note about this process:

- The main idea behind the Python code developed for this project is to 1) understand when the brush has run out of paint, 2) apply more paint to the brush, 3) return to the location where the brush ran out of paint, and 4) continue painting as if the brush never left the canvas.
- Only G01 and G00 commands are used; it just makes the math easier.
- Even for a small four color 200mm by 200mm painting this process may take several hours.
- The swivel axis is not required.
- This process does not detail what to do with the gcode after you have created it. It is up to you to determine how to run it on a three axis cnc machine.
- This method of robotic painting does not use feedback.

The following software programs/add-ons are required:

- Inkscape (open source vector image manipulation program): <https://inkscape.org/en/download/>
- EggBot Inkscape add-on: <https://github.com/evil-mad/EggBot/releases/tag/2.7.1>
 - Wiki: http://wiki.evilmadscientist.com/Hatch_fill
- All to G-Code Converter by CNCToolsWorld.com (a free .dxf or .svg to gcode converter with no cap on file size): https://www.cnctoolsworld.com/application-all_to_g_code_converter_free-2.html
- PyCharm Community Python IDE or your preferred Python IDE: <https://www.jetbrains.com/pycharm/download/#section=windows>
- CAMotics (a gcode viewer): <http://camotics.org/download.html>

Convert to Paint Process:

1. Convert the bitmap image to a vector image
 - a. Open the bitmap image in "Inkscape"
 - b. Use the "Trace Bitmap" feature to convert the bitmap image to a vector image
 - i. Use "Multiple scans" -> "Colors"
 - ii. Adjust the number of scans until you are satisfied with the amount of detail. The number of scans corresponds to the number of paint colors required for the painting.
 - c. Isolate each color into its own layer.
 - i. Select the vector image and select "Object" -> "Ungroup". The image is not separated by color.
 - ii. Remove ("Delete") or combine ("Path" -> "Combine") some scan layers to reduce the number of paint colors required if desired.

- iii. Remove any overlapping layers if desired at this point using the “Path” -> “Difference” feature.
 - iv. Create a new layer for each color.
 - d. For each layer remove the fill color and add a black stroke color.
 - e. Use the “EggBot Contributed” -> “Hatch Fill” feature to hatch each layer.
 - i. The distance between hatch lines should be the width of the paint brush you intend to use. It could be less than the width of your paintbrush if you want paint strokes to overlap.
 - ii. Use different hatch angles on each layer to add style and make it easier to differentiate between layers.
 - iii. Read the wiki, if the “Hatch Fill” feature creates an unintended hatch pattern. The errors are predictable when you know how it works. You may only need to tweak a few parts of your image to get it to work correctly.
 - f. Save each layer as its own “.dxf” file. Use “LWPOLYLINE type of line output” and “inch” base units.
- 2. Convert each dxf layer to gcode in the “All to G-Code Converter”.
 - a. Open the “All to G-Code Converter”.program.
 - b. Use the following g-code settings:

02

Choose your settings:

Starting G-Codes:

G0 {NextPoint}
G1

G-Codes between objects:

G0 {NextPoint}
G1

Ending G-Codes:

- c. Add the correct “.dxf” file location and provide an output file name and location.
 - d. Press the “Convert” button
- 3. Format the g-code to be in a more standard and usable format.
 - a. Open the “CleanGCode” project in the python IDE.
 - b. In the “main.py” script, specify the input and output file directory. Use the output from the “All to G-Code Converter” program as the input for the “CleanGCode” project. Provide a name for the output file.
 - c. Run the “Main” script.
 - d. Repeat steps 3.a to 3.c for each layer
- 4. Enhance the g-code to work for painting applications.
 - a. Open the “ConvertGCode” project in the Python IDE.
 - b. In the “main.py” script, specify the input and output file directory. Use the output from the “CleanGCode” project as the input for the “ConvertGCode” project. Provide a name for the output file.

c. In the “data.py” script, specify the following parameters:

```
# feed rate
feed_rate = 2500 # mm/min

# fourth axis
tool_axial_symmetry = 2 # 0 -> no axial symmetry, 1 -> 180 deg axial symmetry, 2 -> infinite axial symmetry
a_offset = 0 # deg

# max distance to paint before returning for more paint
paint_dist_max = 400

# clean brush after painting specified distance on the canvas
# needed because brush tends to dry out after a period of time
max_clean_brush_dist = 10000

# brush z positions
z_retract = 15
z_plunge = 0

# paint dip
x_paint = -12.5 - 22.5
y_paint = 107.5 - 42.5 - 42.5
z_paint_dip = 1.5
paint_dip_radius = 4.5

# water dip
x_water = -12.5 - 22.5 - 42.5
y_water = 107.5
z_water_dip = 1
water_dip_radius = 10

# towel wipe
x_towel = -12.5 - 22.5 - 21.25
y_towel = 190
z_towel_wipe = 0
towel_wipe_radius = 25
```

Parameter description:

- feed_rate is the speed at which the cnc machine will perform the painting actions
- tool_axial_symmetry controls how the fourth axis will turn relative to the tool direction of travel.
 - 0 implies that the tool has no axial symmetry and that the brush will always turn the same part of the brush in the direction of travel; think angled brush.
 - 1 implies that the tool has 180 degree symmetry and that the fourth axis will either turn the tool to be 0 degrees relative to the direction of travel or 180 degrees relative to the direction of travel; think flat brush.
 - 2 implies that the tool has infinite axial symmetry and therefore no movement is required from the fourth axis; think round brush.
- a_offset is an angle offset that will be applied to the fourth axis. If a_offset is 45 and tool_axial_symmetry is 0, the tool will always travel 45 degrees relative the direction of travel.

- `paint_dist_max` determines how far the brush will draw on the canvas before going back to load the brush with more paint.
- `max_clean_brush_dist` determines how far the brush will draw on the canvas before going back to clean the brush in water, wipe it on a towel, and reapply paint. This prevents paint from drying to the brush.
- `z_retract` is the z-coordinate the tool will be raised to above the canvas
- `z_plunge` is the z-coordinate the tool will plunge to during a movement
- `x_paint` is the x-coordinate of the paint
- `y_paint` is the y-coordinate of the paint
- `z_paint` is the z-coordinate the tool will plunge to when applying paint to the tool
- `paint_dip_radius` is the radius of the circle centered at (`x_paint`, `y_paint`). The program will randomly select a point within this circle when dipping the brush in the paint. If instead the program had the brush go to (`x_paint`, `y_paint`) every time it went to pick up paint then it would create an empty spot in the paint dish and no paint would get applied to the brush.
- `x_water` is the x-coordinate of the water
- `y_water` is the y-coordinate of the water
- `z_water_dip` is the z-coordinate the tool will plunge to when washing the paint brush
- `water_dip_radius` is the radius of the circle that the brush will travel in while cleaning the brush
- `x_towel` is the x-coordinate of the towel
- `y_towel` is the y-coordinate of the towel
- `towel_wipe_radius` is the radius of the circle that the brush will travel in while wiping the brush on the towel

Note: These parameters change based on the type of brush being used, style of painting, type of paint being used, location of paint in the workspace, etc.

- d. Run the “main” script.
5. Review the gcode path in the “CAMotics” gcode viewing software. Note that fourth axis movements cannot be viewed.
6. Repeat step 4 and 5 until you have developed the desired tool path.