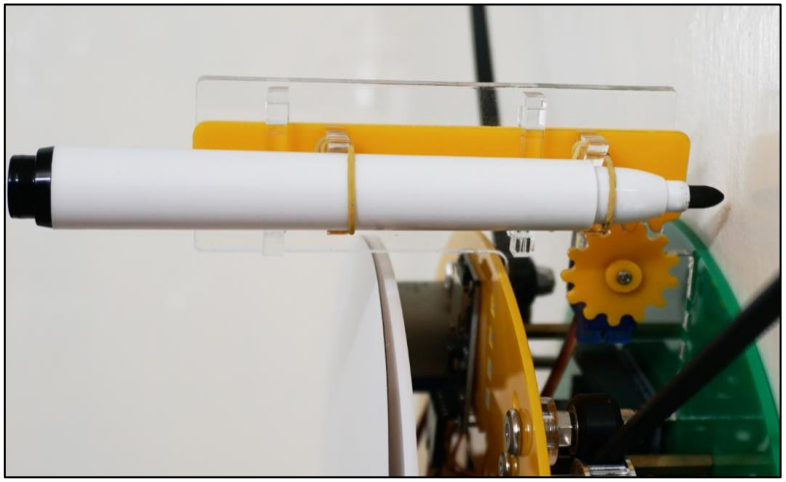
1. HOW TO MAKE DRAWING

WALL DRAWING ROBOT (FUMIK ROBOT)

Author: Nguyen Huu Phuc
Email: baank83@gmail.com
Date: 2022/May

Step 1 **Assembly the pen**

Put the pen to pen driver, tight it by rubber band



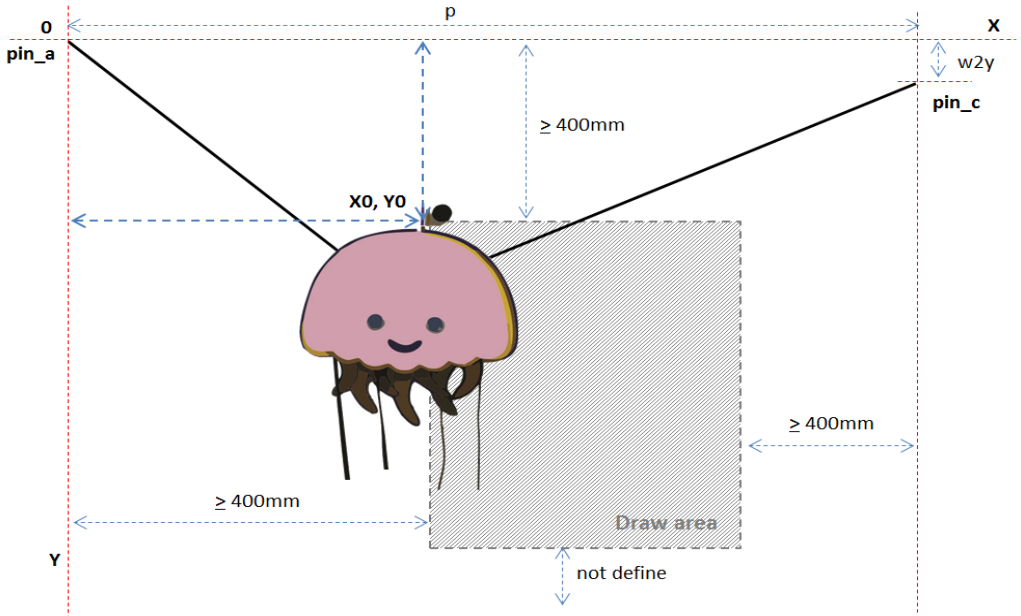
Step 2 Hang robot on 2 nails pinned on wall

Measure distance 2 nails (horizontal direction): p

Measure distance 2 nails (vertical direction): w2y

(Note: if pin_c is higher pin_a, w2y will be minus value)

Measure coordinate of pen: X0, Y0



Put those value to Arduino code

```
p = 2170;    //mm, distance pin_a to pin_c (horizontal direction)
w2y = 450;   //mm, distance pin_a to pin_c (vertical direction)

// zero position of pen must be same as in real model
pen_x0 = 400;  //mm, computer room: 390-400
pen_y0 = 700;  //mm, computer room: 900-700
```

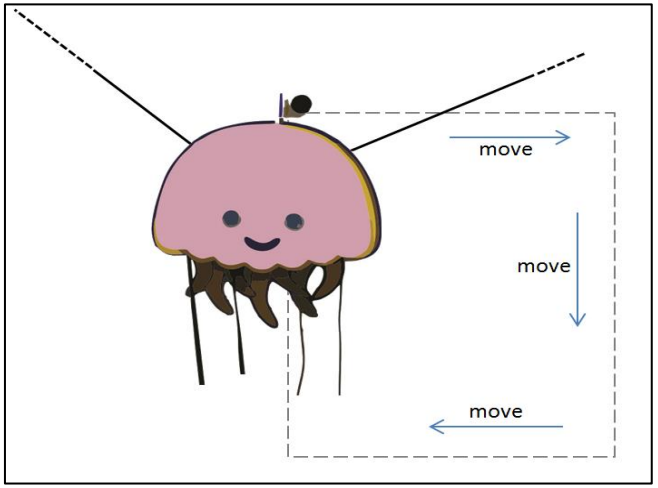
Note: Draw area should keep distance with pin_a, pin_c as above picture (≥ 400mm)

Step 3 **Calibration**

Turn off power of robot

Pull slowly forward/ backward the belt (left/ right) to adjust pen to X0, Y0 location

Copy file calib to sd card, re-name it to "Draw.txt" -> put sd card to robot and power it to start drawing calib line



Robot will draw a rectangle (dot line)

(after finished drawing, robot will return to home position X0, Y0)

Check if the rectangle has exact dimension (width, height)

If not exact, return step 2, check if parameter p, w2y, x0, y0 is exact

If need, adjust *x_ratio, y_ratio* to make drawed rectangle get exact dimension

```
x_ratio = 1.0;  //adjust to change ratio of picture in x_axis
y_ratio = 1.0;  //adjust to change ratio of picture in y_axis
```

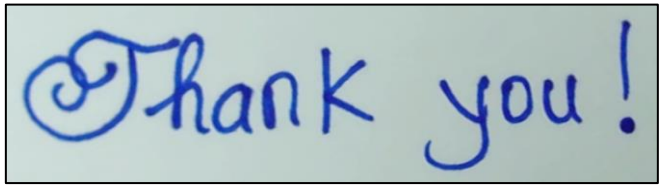
Check if the rectangle is tilt or not

If need, adjust *x_angle, y_angle* to make drawed rectangle to be aligned

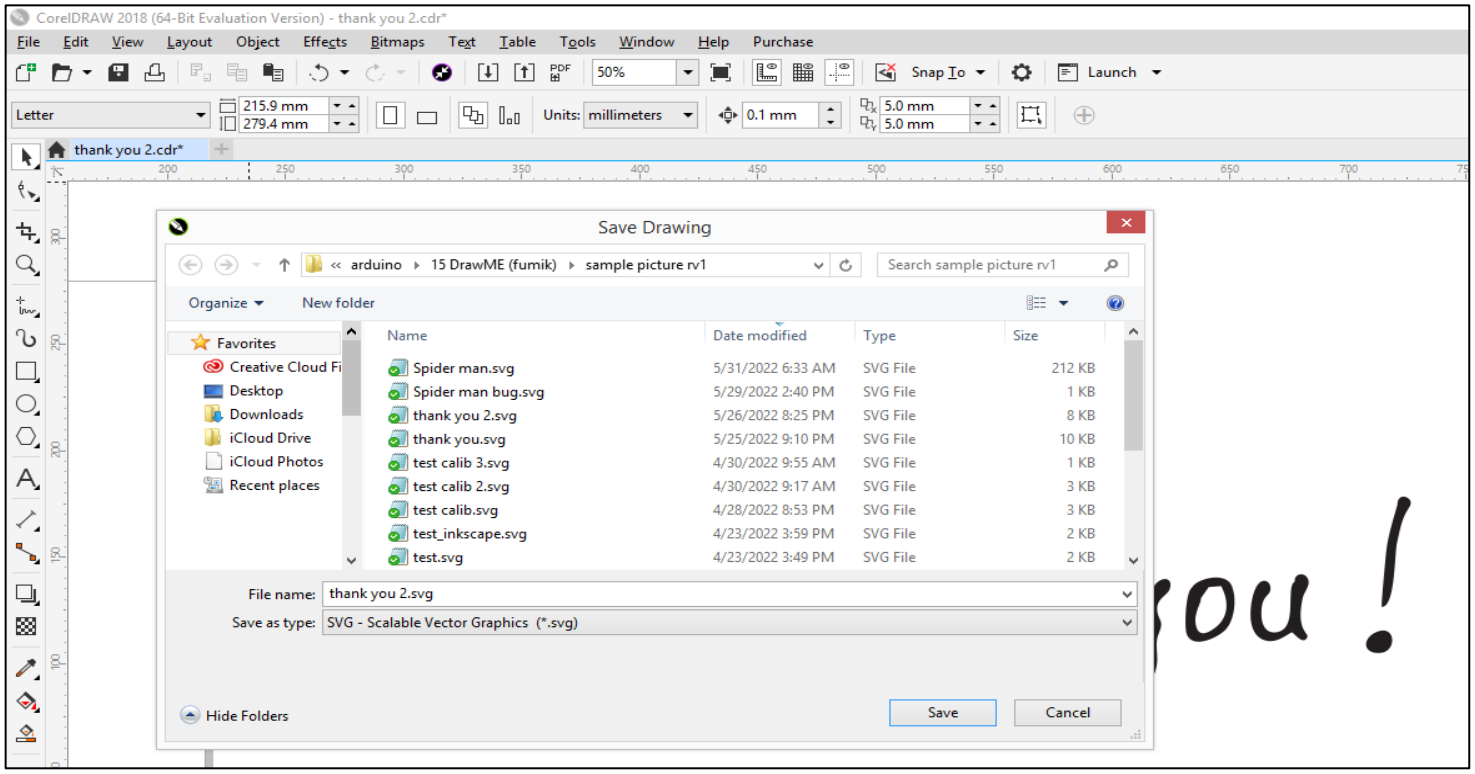
```
x_angle = -1.43; //degree, effect to rotation of picture in x_axis
x_angle = radians(x_angle);

y_angle = -0.36;  //degree, effect to rotation of picture in y_axis
y_angle = radians(y_angle);
```

When the drawed rectangle is acceptable, then robot is ready for other drawings

Step 4 Your Picture need to draw

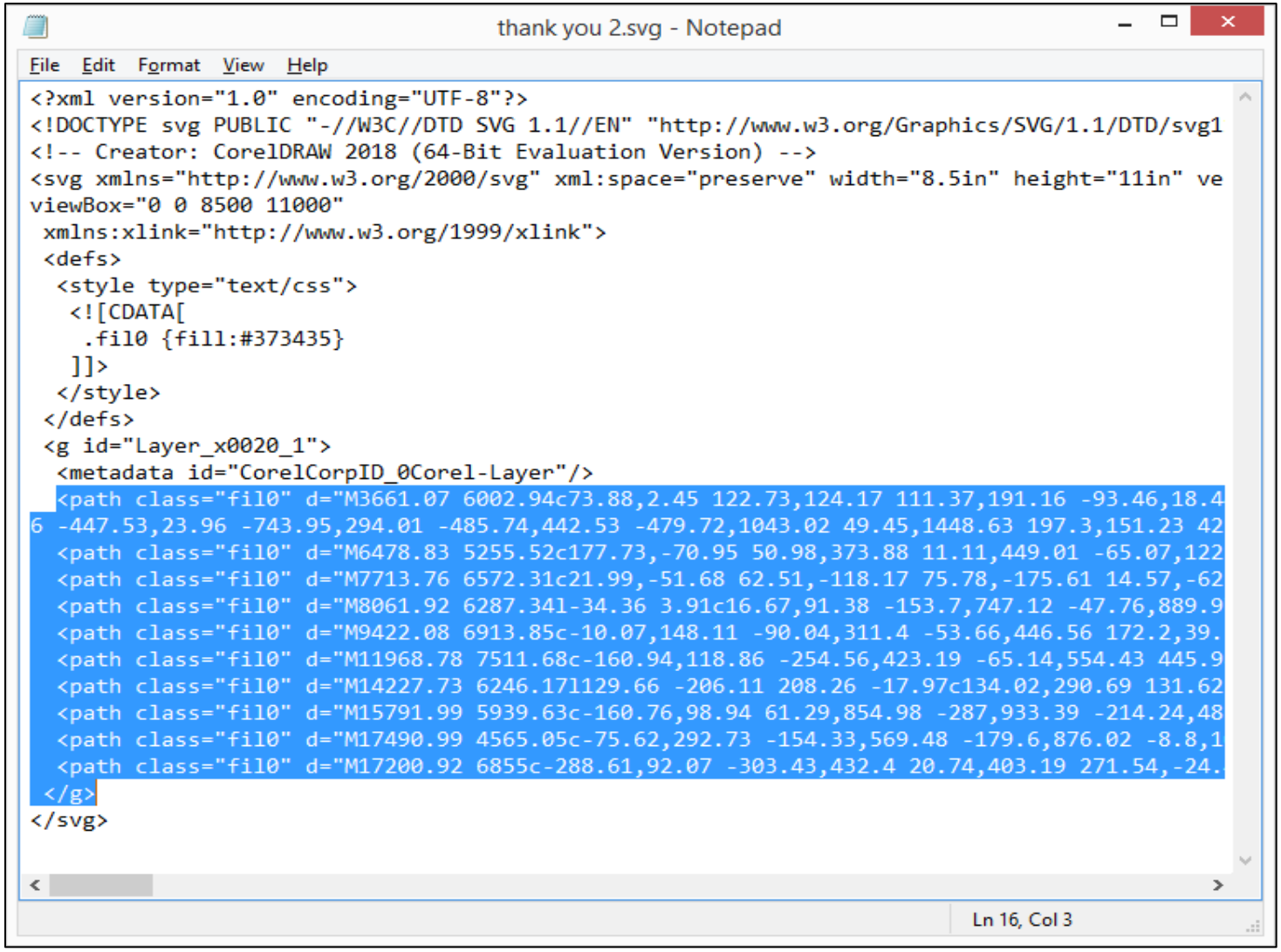


Step 5 Convert picture to vector (*.svg file) by CorelDraw (or any other software support *.svg file)

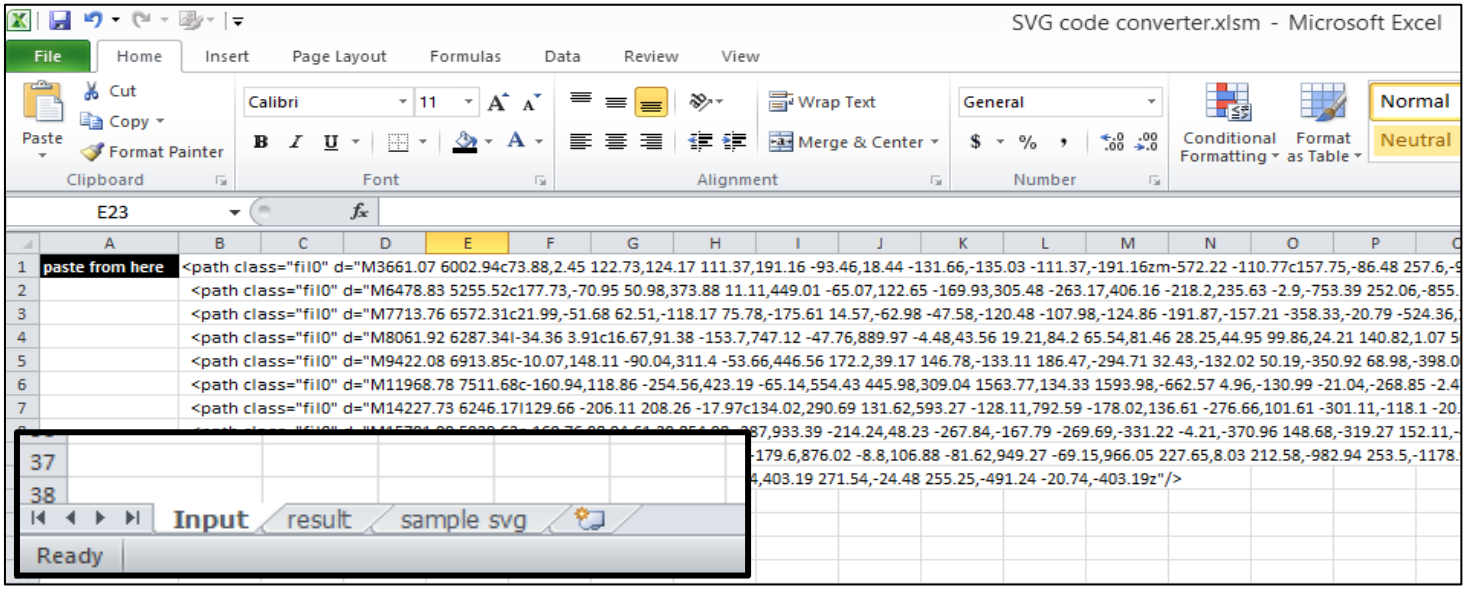

CorelDRAW 2018 (64-Bit Evaluation Version) - thank you 2.cdr*
File Edit View Layout Object Effects Bitmaps Text Table Tools Window Help Purchase
Snap To
Launch
Letter
215.9 mm
279.4 mm
Units: millimeters
0.1 mm
5.0 mm
5.0 mm
thank you 2.cdr*
Save Drawing
arduino ▸ 15 DrawME (fumik) ▸ sample picture rv1
Search sample picture rv1
Organize
New folder
Favorites
Creative Cloud Fi
Desktop
Downloads
iCloud Drive
iCloud Photos
Recent places
Name
Date modified
Type
Size
Spider man.svg 5/31/2022 6:33 AM SVG File 212 KB
Spider man bug.svg 5/29/2022 2:40 PM SVG File 1 KB
thank you 2.svg 5/26/2022 8:25 PM SVG File 8 KB
thank you.svg 5/25/2022 9:10 PM SVG File 10 KB
test calib 3.svg 4/30/2022 9:55 AM SVG File 1 KB
test calib 2.svg 4/30/2022 9:17 AM SVG File 3 KB
test calib.svg 4/28/2022 8:53 PM SVG File 3 KB
test_inkscape.svg 4/23/2022 3:59 PM SVG File 2 KB
test.svg 4/23/2022 3:49 PM SVG File 2 KB
File name: thank you 2.svg
Save as type: SVG - Scalable Vector Graphics (*.svg)
Hide Folders
Save
Cancel
ou !

Step 6 Convert vector file to code for Arduino (by MS Excel) *download here*

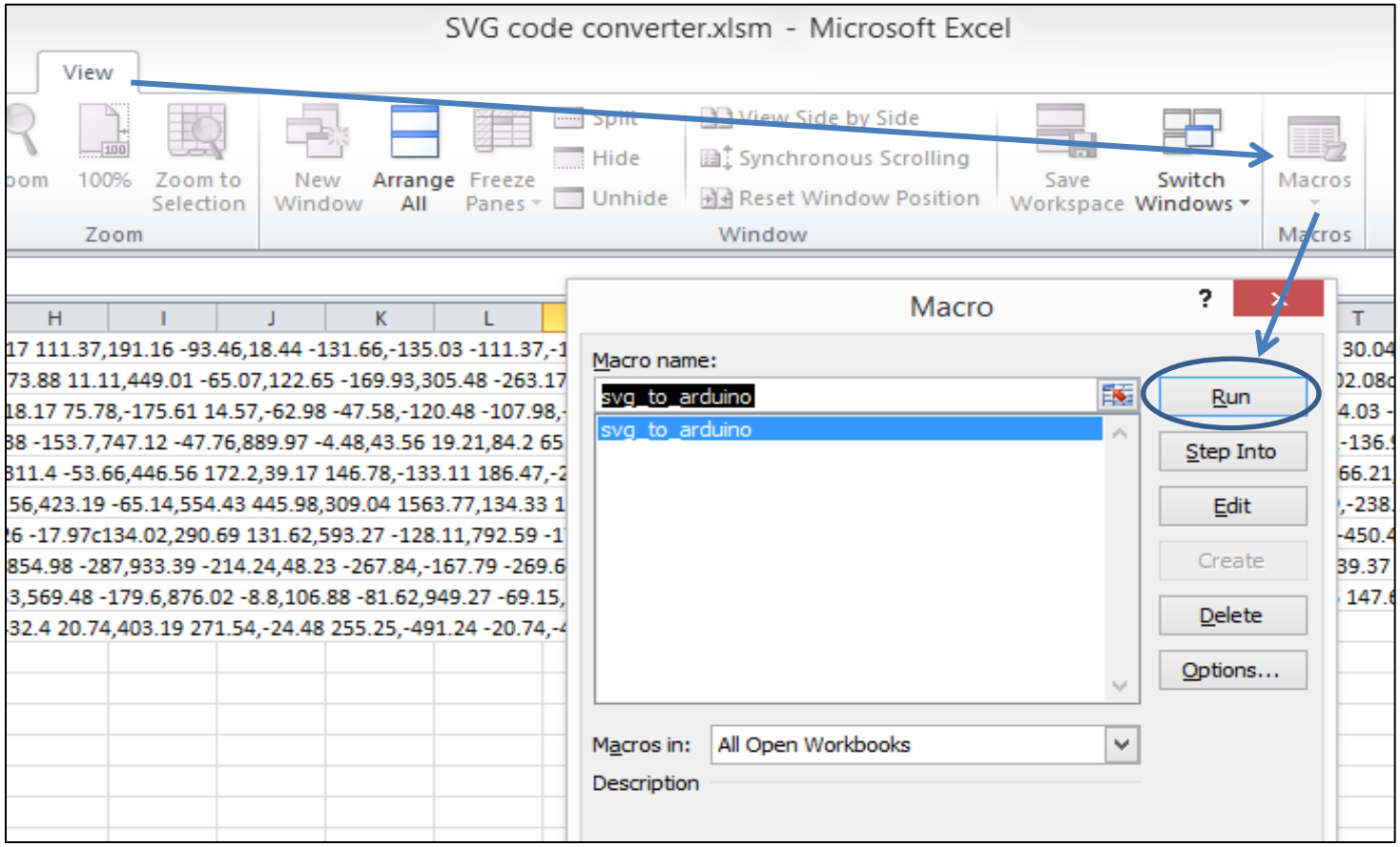
Copy information of line/ curve in *.svg file



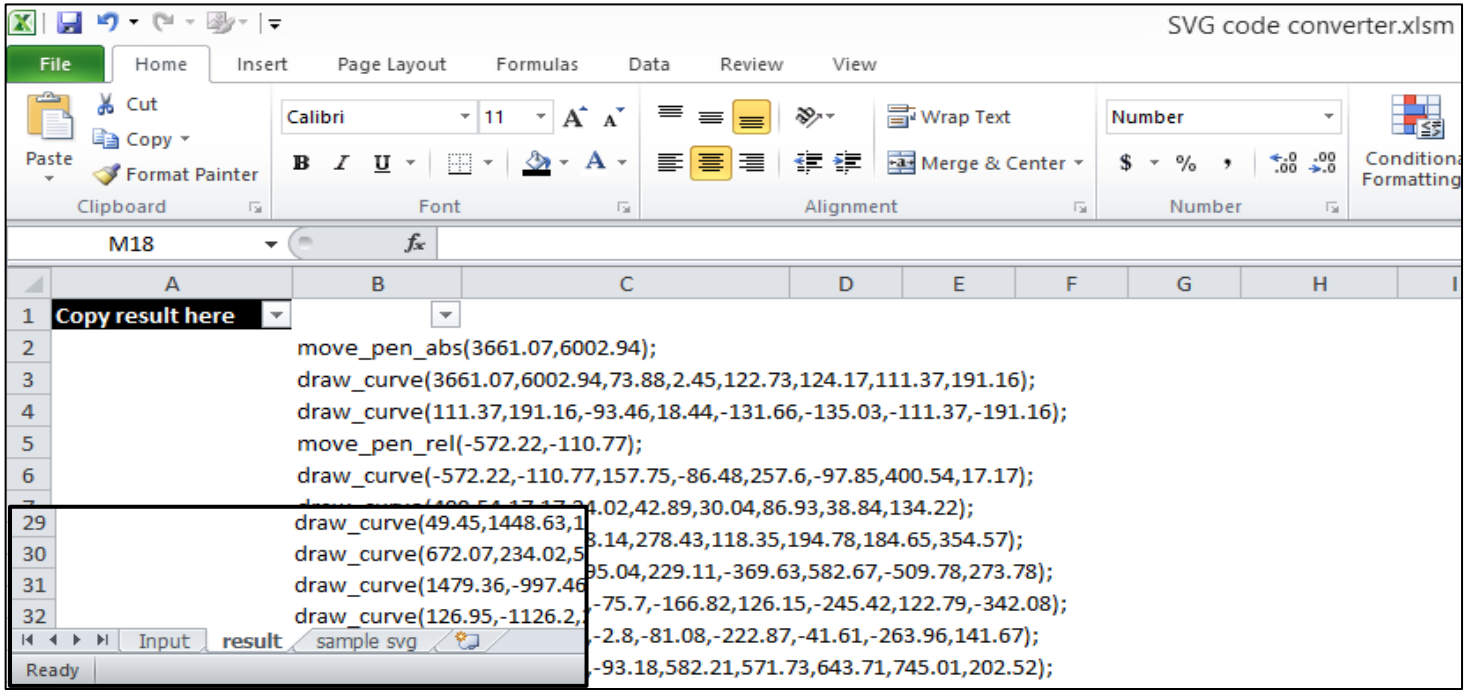
Paste those curve/ line information into converter file "SVG code converter"



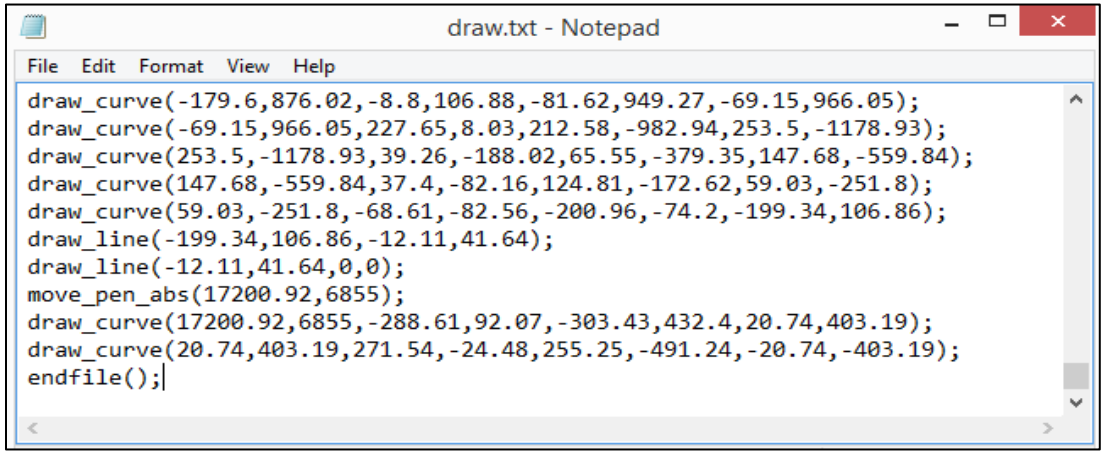
Run "Macro" to convert svg to arduino code



Copy all generated code to text file



At end file, type "endfile();" to inform Arduino about end of file



Step 7 Copy code file "draw.txt" to SD card

Step 8 Plug in SD card to robot and start drawing