# Tests for Gerber File Processing

The following tests will ascertain that Gerber file data is correctly loaded into the Gerber Translator. Each unit test refers to a specific command.

## AS (Axis Select)

Gerber\_AS1.txt contains one line: ASAXBY.

Gerber\_AS2.txt contains one line: ASAYBX.

case "AS":

ProcessAS(commandArgs);

break;

case "FS":

ProcessFS(commandArgs);

break;

case "MI":

ProcessMI(commandArgsArray);

break;

case "MO":

ProcessMO(commandArgs);

break;

case "OF":

ProcessOF(commandArgs);

break;

case "SF":

ProcessSF(commandArgs);

break;

//IMAGE

case "IJ":

ProcessIJ(commandArgs);

break;

case "IN":

this.Image.Name = commandArgs;

break;

case "IO":

ProcessIO(commandArgs);

break;

case "IP":

ProcessIP(commandArgs);

break;

case "IR":

ProcessIR(commandArgs);

break;

case "PF":

this.PlotterFilm = commandArgs;

break;

//LAYER

case "KO":

ProcessKO(commandArgs);

break;

case "LN":

this.Image.Layers[(this.Image.Layers.Count-1)].Name = commandArgs;

break;

case "LP":

ProcessLP(commandArgs);

break;

case "SR":

ProcessSR(commandArgs);

break;

//APERTURE

case "AM":

break;

case "AD":

break;

case "G00":

LastPoint = new Point(GetParam('X', commandArgs), GetParam('Y', commandArgs));

break;

case "G01":

endPoint = new Point(GetParam('X', commandArgs), GetParam('Y', commandArgs));

if (this.ExposureOn)

{

cmp = new Line(LastPoint, endPoint);

}

LastPoint = endPoint;

break;

case "G02":

//Clockwise Circular Interpolation

endPoint = new Point(GetParam('X', commandArgs), GetParam('Y', commandArgs));

centrePoint = new Point(GetParam('I', commandArgs), GetParam('J', commandArgs));

if (this.ExposureOn)

{

cmp = new Circle(this.LastPoint, endPoint, centrePoint, this.CirclesAreSingleQuadrant, true);

}

LastPoint = endPoint;

break;

case "G03":

//CounterClockwise Circular Interpolation

endPoint = new Point(GetParam('X', commandArgs), GetParam('Y', commandArgs));

centrePoint = new Point(GetParam('I', commandArgs), GetParam('J', commandArgs));

if (this.ExposureOn)

{

cmp = new Circle(this.LastPoint, endPoint, centrePoint, this.CirclesAreSingleQuadrant, false);

}

LastPoint = endPoint;

break;

case "G04":

//ignore data block

break;

case "G10":

this.LinearScaleFactor = 10;

break;

case "G11":

this.LinearScaleFactor = 0.1;

break;

case "G12":

this.LinearScaleFactor = 0.01;

break;

case "G36":

this.PolygonAreaFill = true;

break;

case "G37":

this.PolygonAreaFill = false;

break;

case "G54":

//Tool prepare - ignore this.

break;

case "G70":

this.Units = Units.Inches;

break;

case "G71":

this.Units = Units.mm;

break;

case "G74":

this.CirclesAreSingleQuadrant = true;

break;

case "G75":

this.CirclesAreSingleQuadrant = false;

break;

case "G90":

this.CoordinatesAreAbsolute = true;

break;

case "G91":

this.CoordinatesAreAbsolute = false;

break;

case "D01":

this.ExposureOn = true;

break;

case "D1":

this.ExposureOn = true;

break;

case "D02":

this.ExposureOn = false;

break;

case "D2":

this.ExposureOn = false;

break;

case "D03":

break;

case "D3":

break;

case "D10-D999":

break;