

# G-Code Programming

---

## Contents

Overview .....	1
GCodes Summary Table .....	2
GCodes In Detail .....	3
Comments .....	3
Coordinate System .....	3
Cutter Compensation.....	3
Feedrate .....	3
Leadin .....	3
Linear Motion .....	4
Scale .....	4
Surface Intersection.....	4
Tabs .....	5
Torch .....	5
Torch On/Off .....	5
Torch Position/Motion.....	5
Sample G-Code: .....	6
Using the G-Code Parser in PypeServer .....	7
Validating and Saving the G-Code .....	7
Finding Errors.....	7
The JSON format.....	8

## 1 Overview

The pipe cutting machine can be controlled directly using GCode cuts. These cuts are programmed like traditional machining G-Code programs, though most commands are more descriptive.

GCode cuts are typically used when the motion of a cut is defined in CAD as a shape (instead of parameters), or the cut is not easily defined mathematically. Gcodes sources include:

- Cuts imported from CAD exported data where the data is a set of points.
  - DSTV files, which includes Tekla .nc1 files
  - .DXF, DWG, IGES, STEP (customers can create importers for these. See The JSON Format for more information)
- Cuts programmed manually by the operator by entering data into the PypeServer cut editor.
- Cuts created programmatically by you or a 3<sup>rd</sup> party. (Typically created in JSON format).

## 2 GCodes Summary Table

Function	GCodes	Desc	Use in header	Use while cutting
Comments	// comment	Comments can be alone on a line, or can follow a command	X	X
Coordinate System	COORDS_CART (default)	Coordinates input are Cartesian Right-Handed system	X	
Cutter Comp	COMP_OFF (default) COMP_LEFT COMP_RIGHT COMP_AUTO	Comp determines whether the torch will be on the left, right, or center of the cut line.	X	
FeedRate	F <real num>	Same function as in PypeServer	X	X
LeadIn	LEADIN_OFF (default) LEADIN_ON	Have PypeServer generate a leadin (and if set in Torch Settings, a leadout)	X	
Linear Motion	G01 <x,y,z> G01 <x,y,z,x2,y2,z2> G02 <x,y,z,u,v>	Linear motions		X
Scale	SCALE <x,y,z>	Set the scale for G01 & G02 numeric input	X	X
Surface Intersection	SURF_ID SURF_OD (default)	Defines the primary cut surface (see details)	X	
Tabs	TABS <integer num> (default = 0)	Same function as in PypeServer	X	
Torch	T_PLASMA (default) T_GAS T_ETCH T_MARK	The torch to use for the cut	X	
Torch Motion	TUV <u,v>	Set Torch motion without linear motion	X	X
Torch OnOff	TORCH_ON TORCH_OFF	Turn torch on and off		X

## 3 GCodes In Detail

Default values apply if nothing is set in the GCode.

### 3.1 Comments

- `//<comment>`  
Comments can go on a line along, or can follow a command

### 3.2 Coordinate System

- `COORDS_CART` (default)  
Coordinates input are Cartesian Right-Handed system. You can change the coord system with scale. E.g. change to left-handed, set Scale to 1, 1, -1 (reverse Z).
- `COORDS_CYL`  
Cylindrical coordinates (Y down the pipe, X rotation).  
Currently not supported. Contact PypeServer if you need this feature.

### 3.3 Cutter Compensation

- `COMP_OFF` (default)
- `COMP_LEFT`
- `COMP_RIGHT`
- `COMP_AUTO` – the system will use the direction of the cut as it is input to determine the cut side.

NOTE: PypeServer always keeps the cutting torch on the torch swirl “climb mill” side. So as you change from `COMP_LEFT` to `COMP_RIGHT` the system will reverse the cut direction of the cut to keep the swirl on the correct side for optimal cutting.

### 3.4 Feadrates

- `F <real number greater than zero>`

### 3.5 Leadin

- `LEADIN_OFF` (default)
- `LEADIN_ON`

#### NOTES

- If you have a leadin built into the G-Code, make sure this is set to `LEADIN_OFF`
- Leadins may not always work correctly with GCode. Visually check the lead-ins (and outs) for each cut.
- Using leadin will set your torch pierce angle to the torch settings, so if you are using a static UV torch setting, make sure to set TUV before beginning your cut.

### 3.6 Linear Motion

These apply to COORDS\_CART coordinate system

- G01 <x,y,z>  
X provides the distance down the pipe. Y and Z provide a vector from the centerline of the pipe that is intersected to the defined surface intersection of the pipe.
  - Y is up, Z is towards the viewer (unless SCALE changes the meanings)
  - The YZ point does not need to be at the surface intersection of the pipe, as the vector will be lengthened or shortened to the surface intersection.
  - The current U and V torch angle settings are applied at this intersection point.
- G01 <x,y,z,x2,y2,z2>  
The two xyz points define a line segment in 3D Cartesian space. This segment is used to intersect the inner and outer surfaces of the pipe. Note that the segment points do not need to be at the ID and/or OD of the pipe, as the segment will be extended in both directions to make the intersection.
  - In effect there is no “driving” surface, as both surface intersections are defined by the two points.
- G02 <x,y,z,u,v>  
This is the same as G01, except with the torch U and V angles updated.

### 3.7 Scale

- SCALE <x,y,z>  
Examples:
  - SCALE 1,1,1 //this is the default (nothing changes)
  - SCALE 1,1,-1 //reverse Z
  - SCALE 0.03937, 0.03937, 0.03937 //data is being input in metric,

### 3.8 Surface Intersection

- SURF\_ID
- SURF\_OD

The G01 and G02 positions do not always define absolute points of intersection on a pipe. This is because a pipe diameter can (and often does) vary from the “ideal” imported or manually design. Surface Intersection defines the surface on which motion points will stay fixed when torch angles are applied. Here are the uses per different input definitions for COORDS\_CART:

- G01 <x,y,z> and G02 <x,y,z,u,v>  
The YZ is used to determine a vector out from the centerline of the pipe. This vector is intersected with the specified Surface Intersection. The torch U and V angles (TUV) are then applied from that point.
- G01 <x,y,z,x2,y2,z2>  
Surface Intersection is unused. See Linear Motion for this input type.

### 3.9 Tabs

- TABS <integer number>
  - Default is zero
  - If TORCH\_OFF is used before the end of the cut, TABS will be turned off (set to zero).

### 3.10 Torch

- T\_PLASMA
- T\_GAS
- T\_MARK
- T\_ETCH

NOTES:

- Your machine must be configured with Mark or Etch to use these settings

### 3.11 Torch On/Off

- TORCH\_ON
- TORCH\_OFF

NOTES:

- If TORCH\_OFF is used before the end of the cut, TABS will be turned off (set to zero). This is because TABS is a specific mode of controlling the torch on and off during the cut.
- If Torch ON/OFF is not set at the beginning of cut motion, the system will turn the torch is to be on at the first step.
- If LEADIN is set to on, then torch on will begin at the beginning of the leadin
- If you are using a GAS torch, the system will pause and wait at the first step for the operator to hit go after the torch is ignited and the hole is pierced.

### 3.12 Torch Position/Motion

- TUV <u,v>
  - The u parameter defines the torch angle that lays the torch down along the length of the pipe.
  - The v parameter defines the torch angle that lays the torch out across the pipe

## 4 Sample G-Code:

```
COORDS_CART
COMP_RIGHT
LEADIN_ON
TABS 0
T_PLASMA
SURF_OD
F 40.00
TUV 0.00, 0.00 //this will stay at this setting the entire cut
G01 -0.751, 0.950, 0.000
G01 -0.785, 0.950, 0.000
G01 -0.603, 0.915, -0.254
G01 -0.513, 0.806, -0.504
G01 -0.378, 0.671, -0.673
G01 -0.277, 0.407, -0.858
G01 -0.069, 0.180, -0.933
TORCH_OFF //tab start
G01 0.173, -0.100, -0.945
G01 0.416, -0.356, -0.881
TORCH_ON //tab stop
G01 0.585, -0.574, -0.757
G01 0.705, -0.787, -0.532
G01 0.809, -0.889, -0.336
G01 0.918, -0.950, -0.026
G01 0.861, -0.908, 0.280
G01 0.747, -0.780, 0.543
G01 0.598, -0.610, 0.728
G01 0.357, -0.349, 0.884
G01 0.128, -0.088, 0.946
G01 -0.080, 0.150, 0.938
G01 -0.317, 0.408, 0.858
G01 -0.440, 0.563, 0.766
G01 -0.643, 0.765, 0.563
G01 -0.691, 0.902, 0.297
G01 -0.751, 0.950, 0.000
```

The G-Code parser will appear in the PyeServer Part editor when you are creating or editing a G-Code cut.

Validate & Save validates the G-Code (and posts errors if there are any), and (whether the data is good or not) saves the G-Code to the server. If "Auto-Validate/Save is checked, this will be done for you after 2 seconds of idle time when inputting data.

The GCode Parser will parse the gcode file and provide information on parsing errors. Clicking on the error items will take you to the error in the file.

## 6 The JSON format

PypeServer saves GCode in a [JavaScript Object Notion](#) (JSON) format. This is an industry standard that is an easier format than .xml for less structured data management. This format is recommended if you wish to create your own GCode cuts programmatically. Contact PypeServer to discuss programming requirements and the C# classes defining the JSON Part, Cut and GCode step objects.