

Woodmeister 사용설명서

Python으로 만든 아주 쉬운 목공용 CAD/CAM 프로그램



2024 May 22

yskim57@gmail.com

목차

[1. Woodmeister는 무엇인가요? 4](#_Toc172386294)

[1.1. Woodmeister 의 특징은? 4](#_Toc172386295)

[1.2. Woodmeister의 설치 4](#_Toc172386296)

[2. Woodmeister 의 구성 5](#_Toc172386297)

[2.1. 좌표축 및 가공면 5](#_Toc172386298)

[2.2. Top Window 6](#_Toc172386299)

[2.3. 디자인 순서 7](#_Toc172386300)

[2.4. 디자인 시작 전: 환경설정 8](#_Toc172386301)

[3. 디자인 시작 9](#_Toc172386302)

[3.1. 모재(BASE)의 지정 10](#_Toc172386303)

[3.2. 디자인 요소의 추가 10](#_Toc172386304)

[3.2.1. 주요 디자인 요소: 홀(Hole) 과 기둥(Post) 11](#_Toc172386305)

[3.2.2. 디자인 요소의 변경/복제/삭제/되돌리기 12](#_Toc172386306)

[3.2.3. 디자인 요소의 대칭/회전 13](#_Toc172386307)

[3.2.4. Offset 15](#_Toc172386308)

[4. 디자인 요소 살펴보기 17](#_Toc172386309)

[4.1. 원형홀(Circular Hole)/원형기둥(Circular Post) 17](#_Toc172386310)

[4.2. 다각형홀(Polygon Hole)/다각형기둥(Polygon Post) 20](#_Toc172386311)

[4.3. 사각형홀(Rect\_hole)/사각형기둥(Rect\_Post) 20](#_Toc172386312)

[4.4. 직선(Line)과 원호(Arc) 21](#_Toc172386313)

[4.5. 다중홈(Multi Trench) 23](#_Toc172386314)

[4.6. 옆면 다중홈(Side Trench) : 23](#_Toc172386315)

[4.7. 드릴홀(Drill Hole) 24](#_Toc172386316)

[4.8. 임의형상(자유곡선, CURVE) 25](#_Toc172386317)

[4.9. 참조선(REF\_LINE) 26](#_Toc172386318)

[5. 라이브러리/문자열/특수도형/사용자 정의 프로그램 27](#_Toc172386319)

[5.1. Import 27](#_Toc172386320)

[5.2. 특수 도형/디자인 생성 27](#_Toc172386321)

[5.3. 문자열 생성 27](#_Toc172386322)

[5.4. 사용자 프로그램 수행 28](#_Toc172386323)

[6. G-code 생성 및 관련 버튼 28](#_Toc172386324)

[6.1. G-code 생성 29](#_Toc172386325)

[6.2. G-code 보여주기 30](#_Toc172386326)

[6.3. G-code 복제/합성 30](#_Toc172386327)

[6.4. 문자열 생성 – StickFont 사용 31](#_Toc172386328)

[7. 측정 및 접선 32](#_Toc172386329)

[8. Tool table 33](#_Toc172386330)

[8.1. 제한사항 34](#_Toc172386331)

[8.2. Tool Table의 항목 34](#_Toc172386332)

[9. Group 작업 35](#_Toc172386333)

[9.1. 복사 후 붙여넣기 35](#_Toc172386334)

[9.2. 이동 36](#_Toc172386335)

[9.3. 선대칭 36](#_Toc172386336)

[9.4. 점대칭 36](#_Toc172386337)

[9.5. 요소 삭제 36](#_Toc172386338)

[9.6. 요소 회전 36](#_Toc172386339)

[9.7. Export 36](#_Toc172386340)

[9.8. 조건 부합 요소 변경 36](#_Toc172386341)

[10. Keyboard 와 Mouse 39](#_Toc172386342)

[11. Advanced Topic 40](#_Toc172386343)

[11.1. Object 형식 40](#_Toc172386344)

[11.1.1. 모재(BASE) 40](#_Toc172386345)

[11.1.2. 원형홀(Circular Hole) 41](#_Toc172386346)

[11.1.3. 다각형홀(Polygon Hole) 41](#_Toc172386347)

[11.1.4. 사각형홀(Rectangular Hole) 42](#_Toc172386348)

[11.1.5. 원형기둥(Circular Post) 43](#_Toc172386349)

[11.1.6. 다각형기둥(Polygon Post) 43](#_Toc172386350)

[11.1.7. 사각기둥(Rectangular Post) 43](#_Toc172386351)

[11.1.8. 직선(Line) 44](#_Toc172386352)

[11.1.9. 원호(Arc) 44](#_Toc172386353)

[11.1.10. 상면다중홈(MULTI\_TRENCH) 45](#_Toc172386354)

[11.1.11. 옆면다중홈(SIDE\_TRENCH) 45](#_Toc172386355)

[11.1.12. 드릴홀(Drill-Hole) 46](#_Toc172386356)

[11.1.13. 임의형상(Curve) 46](#_Toc172386357)

[11.1.14. 참조선(Ref-Line) 46](#_Toc172386358)

[11.2. 디자인 File의 구조 47](#_Toc172386359)

[11.3. 문서편집기를 이용한 디자인 생성 49](#_Toc172386360)

[11.4. Python 프로그램을 이용한 디자인 생성 51](#_Toc172386361)

[11.5. Point sequence 기반 사용자 정의 자유 형상 구현 – 57](#_Toc172386362)

[11.5.1. Point sequence 생성 - Script command 활용 58](#_Toc172386363)

[11.5.2. Point sequence file의 생성 – 직접 편집/프로그램에 의한 생성 63](#_Toc172386364)

[11.5.3. Point sequence file의 생성 - 접선 경로의 생성 66](#_Toc172386365)

[부록. Video Contents 71](#_Toc172386366)

# Woodmeister는 무엇인가요?

Woodmeister는 목공을 위하여 개발한 전용 CAD 프로그램입니다. 아마추어라도 단시간 내에 사용방법을 습득하여 목표로 하는 디자인을 완성하고 CNC 가공기를 통하여 직접 제작을 할 수 있도록 Tool 경로까지 생성해주는 CAD와 CAM 일체형 프로그램입니다

## Woodmeister 의 특징은?

Woodmeister는 목재 가공에 특화된 CAD와 CAM을 일체화한 전용 프로그램으로써 CAD 및 CAM 의 기능들은 아래와 같이 정리할 수 있습니다.

CAD :

* + 목재 가공에 필요한 요소(표면, Hole, Post, 선 등)의 가공을 위한 간소화된 설계 지원
  + Tool의 교체를 통하여 디자인 요소에 대한 Trim 가공을 할 수 있는 기능 제공 (Post-Processing)
  + 설계는 모재(Base)의 전개도 위에 각종 디자인 요소를 배치하고 이의 모양을 정의하는 과정임

CAM:

* + 설계도면에 기반한 CNC용 G-code 생성
  + Tool 의 특성을 적용한 맞춤형 G-code 생성

## Woodmeister의 설치

Woodmeister의 실행 파일 및 관련정보는 아래의 링크(Link)에서 구할 수 있다.



[https://github.com/iamyskim57/Python\_woodworking](https://github.com/iamyskim57/python_woodworking)

# Woodmeister 의 구성

Woodmeister는 아래와 같은 몇 개의 Window로 구성이 된다.

* Top Window
  + 디자인을 시작하거나 기존의 디자인을 읽어 들여 편집할 수 있도록 하는 주 화면
* 디자인 Window
  + 모재의 형태 및 디자인 요소들을 배치하는 화면
* Feature Window
  + 디자인 요소의 상세한 내용을 표기/변경하도록 하여주는 Sub-window
* Measurement Window
  + 2 개의 요소간의 관계에 기반한 몇가지 행위를 할 수 있는 Sub-window
    - 두개 요소간의 거리,
    - 두개의 원을 지나는 접선의 생성,
    - 시작하는 접선과 이에 접하는 원을 시작으로 하는 원과 이의 접선으로 이루어지는 경로를 추출하는 기능
* Tool Table
  + - 사용하는 가공용 Tool의 특성을 편집하는 Sub-window
* 사용환경 설정 Window
  + - 디자인 data, G-code, Library 등의 data를 저장하는 기본 folder 를 지정하고 기타 운용 환경을 설정하는 Sub-window

## 좌표축 및 가공면

가공면은 가공이 이루어지는 면으로써 Tool이 직접 도달하여 가공이 이루어지는 평면이다. 일반적으로 이는 아래와 같은 좌표시스템에서 XY 평면을 기본으로 한다.

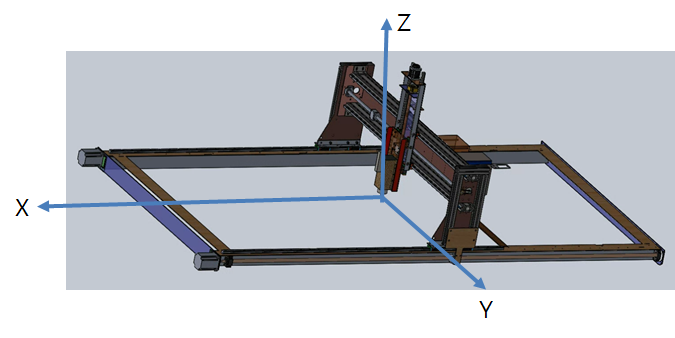
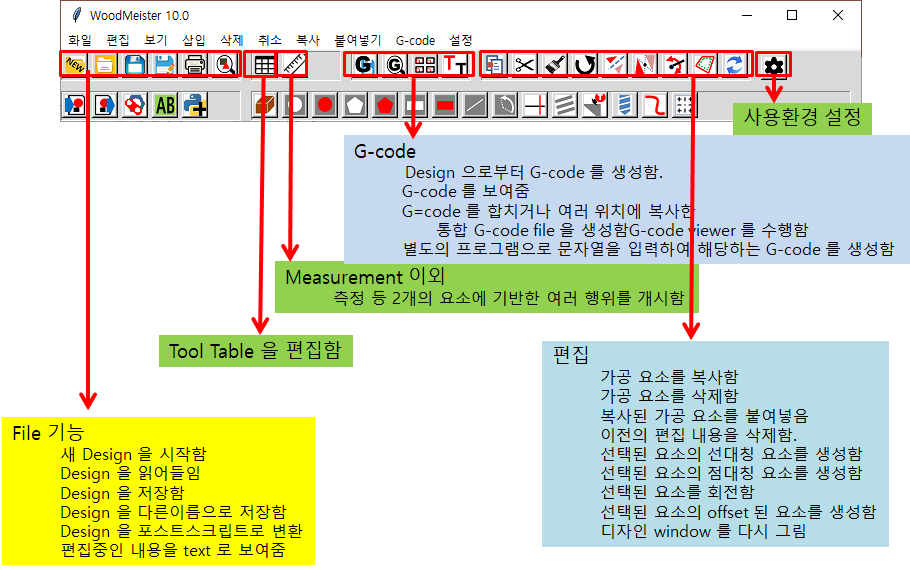


그림 1. 좌표축과 가공면

## Top Window



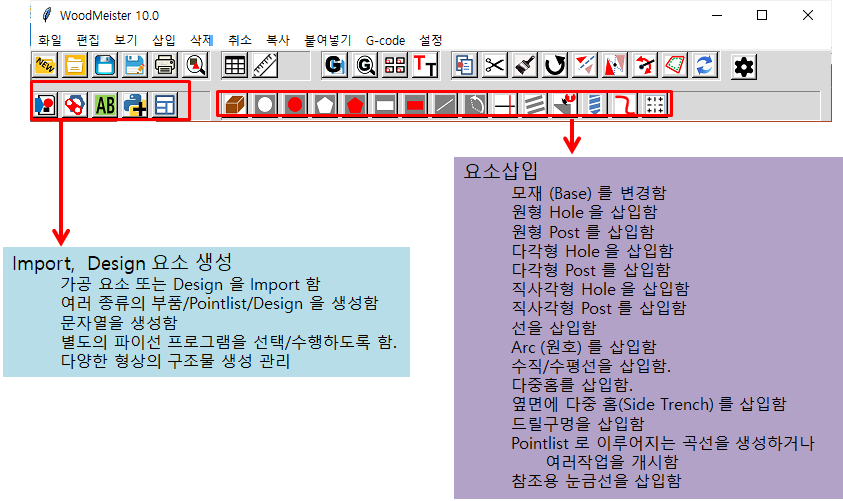


그림 2. Top Window

Top window는 프로그램의 수행을 시작하면 보여주는 Window로 모든 Sub-window를 불러내거나 디자인에 대한 모든 제어를 할 수 있는 Window이다. 각 버튼의 기능들은 그림 1을 참조하기 바란다. 이들 버튼의 기능은 메뉴바를 통하여서도 수행할 수 있다. 메뉴바의 주메뉴를 클릭하면 이에 종속되는 하위의 메뉴(Sub-menu)가 표기되어 이들 중 하나를 선택하여 수행할 수 있다.

## 디자인 순서

Woodmeister 디자인은 몇개의 단계를 거친다. 그 순서는 아래와 같이 정할 수 있다.

1. 환경설정 :  또는 설정
2. 신규디자인 : , 시작화일->새로만들기
3. 모재 정의:  또는 ‘삽입->모재’
4. 디자인 요소 삽입: ,

Library, 문자열, …

1. 디자인 요소 편집: 
2. 디자인 저장:  또는 ‘화일->다른 이름’으로 저장
3. 가공 Data (G-code) 생성 :  또는 ‘G-code->생성/저장’

최종적으로 나오는 G-code는 CNC 기계를 작동하기 위한 code 로써 CNC 기기에 따라 약간씩 다른 문법을 가질 수 있으며 Woodmeister 는 GRBL11을 기본으로 하고 있다.

## 디자인 시작 전: 환경설정

본격적인 디자인을 시작하기 전에 가장 먼저 할 일은 환경을 설정하는 것으로써 작업할 내용을 저장할 Folder, Library Folder 등 필요한 사항들을 설정한다. 아래의 그림은 환경설정 Window 이다. 설정한 환경을 저장하고 종료한 후 본격적인 디자인을 시작한다.

* 디자인폴더: 디자인을 저장하는 Folder
* G-코드 폴더 : 생성한 G-code 를 저장하는 Folder
* 라이브러리 : Import/export 를 통하여 디자인 요소를 읽어오거나 저장하는 Folder
* 툴표 : Tool Table 의 위치
* 언어 : 영어 또는 한국어 중 선택
* 원호표현 : 원호표현의 방식을 정함.
* Controller : CNC 기기의 controller를 지정함.

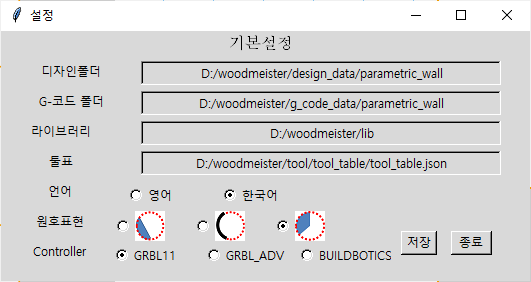


그림 3. 환경설정 Window

# 디자인 시작

Woodmeister를 실행하면 그림 2 와 같은 Top window 가 화면에 표기된다. 새로운 디자인을 시작하기 위하여 ‘화일->새로만들기’ 또는  버튼을 누른다. 이후 디자인 Window 가 아래의 그림과 같이 떠오른다.

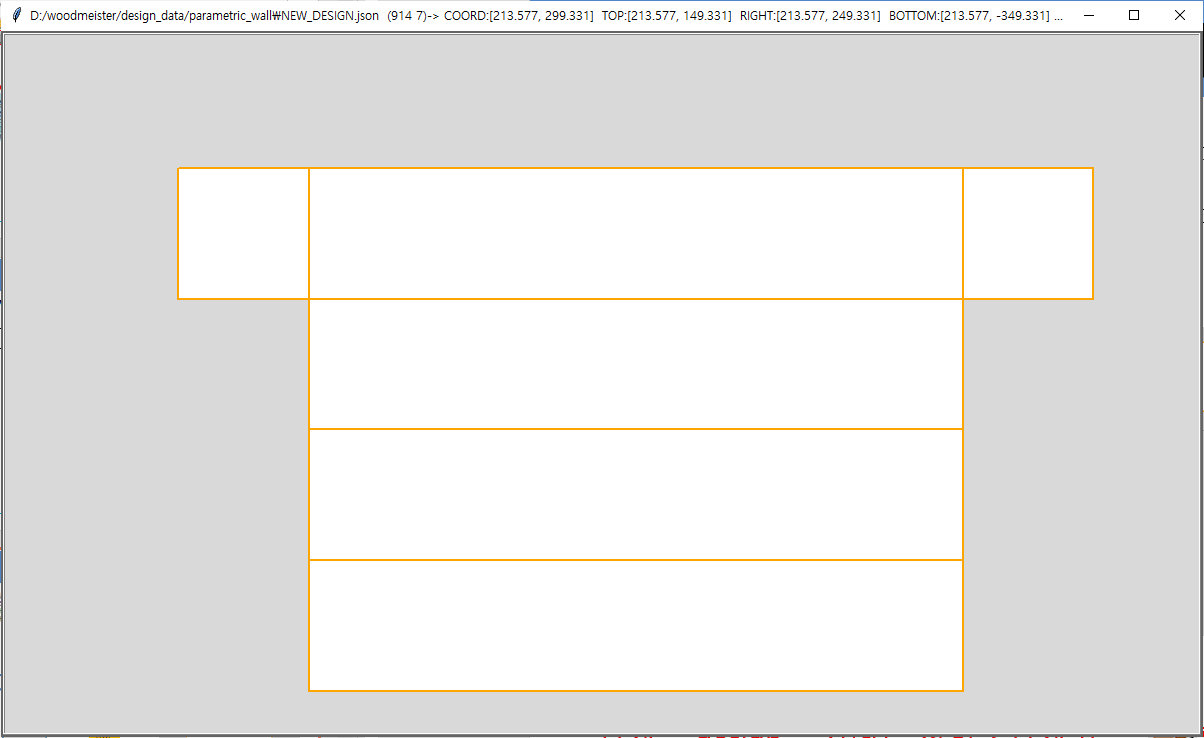


그림 4. 디자인 Window

디자인 Window에는 육면체를 전개한 전개도가 표시된다. 이의 각각의 면은 아래의 그림 5와 같이 지정된다. 전개도를 접으면 육면체인 모재의 형태가 되는 개념이다. 디자인은 이 육면체의 각각의 면에 디자인 요소를 배치하는 것으로 간주할 수 있다.

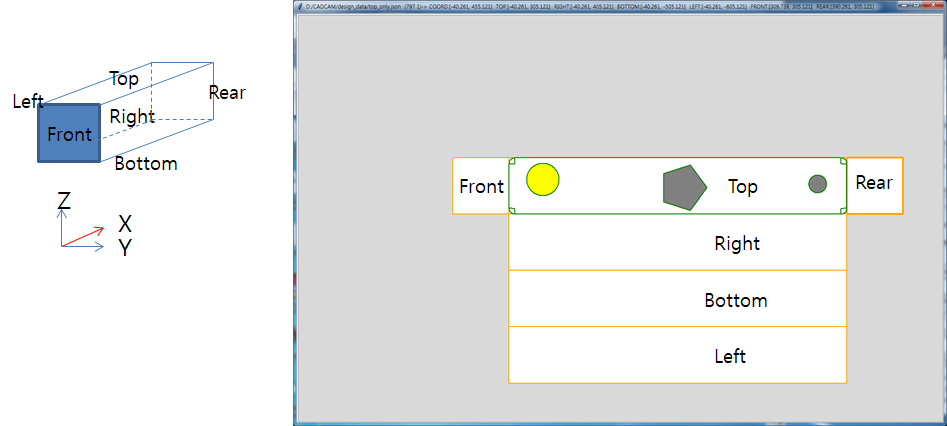


그림 5. 디자인 window 와 모재의 관계

이때 설정되는 모재는 Woodmeister가 기억하고 있는 가장 최근에 사용하였던 모재로 결정된다.

## 모재(BASE)의 지정

모재의 형태는 항상 바꿀 수 있으며 ‘삽입->모재’ 또는  버튼을 눌러서 그 형태를 바꿀 수 있다. 이와 같이 하면 아래와 같이 모재의 특성을 표현하는 Feature window 가 떠오른다. 모재는 높이, 길이, 폭의 3개 수치로 표현할 수 있다. 이들 각각의 수치는 mm 단위의 숫자이며 각각의 입력난에 수치를 입력한 후 ‘추가’ 버튼을 눌러 새로이 모재를 지정한다. 모재를 추가하면 디자인의 중간이라 하더라도 모재를 새로 적용하여 디자인 화면을 새로이 만들어 준다. **모재를 변경하기 위해서는 항시 ‘추가’ 버튼을 사용하여야 한다.**

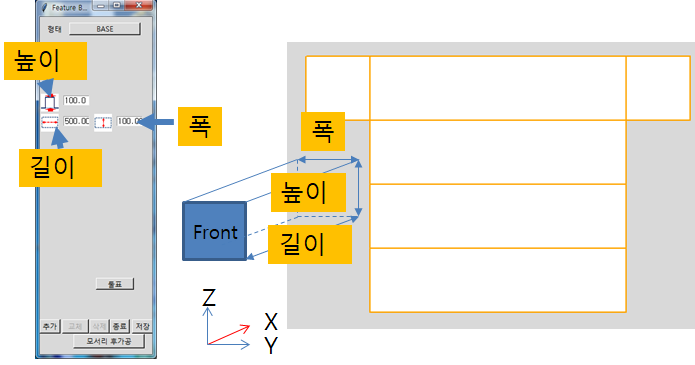


그림 6. 모재의 형상 지정

## 디자인 요소의 추가

모재가 지정된 후에 디자인 요소들을 삽입할 수 있다. 이들의 삽입에는 아래의 버튼들을 이용할 수 있다. 또한 상단 메뉴바에서 ‘삽입-> 원형홀’과 같은 방식으로 삽입할 수도 있다.



그림 7. 디자인 요소 추가를 위한 버튼들

그림 7의 각각의 버튼들은 왼쪽부터 모재(Base), 원형홀(Circular Hole), 원형포스트(Circular Post), 다각형홀(Polygon Hole), 다각형포스트(Polygon Post), 직사각형홀(Rectangular Hole), 직사각형포스트(Rectangular Post), 직선, 원호, 수평/수직선, 다중홈, 옆면다중홈, 드릴홀, 자유곡선, 참조선을 삽입하고자 할 때 사용한다. 이들 버튼들을 누르면 해당 요소에 맞는 Feature Window 가 표기되며 이를 통하여 해당 요소의 상세한 내용을 편집할 수 있다.



그림 8. Feature Window의 예. Feature window는 디자인 요소에 따라 서로 다른 특성을 표기하고 변경을 할 수 있도록 해준다. 이를 통하여 가공하고자 하는 요소의 특성을 정확히 지정할 수 있다.

Feature window에서 디자인 요소의 특성을 변경하고자 할 때는 변경된 값을 모두 입력한 후 반드시 추가 또는 교체 버튼 중 활성화 된 버튼을 눌러 변경 사항을 반영하여 주어야 한다.

### 주요 디자인 요소: 홀(Hole) 과 기둥(Post)

홀과 기둥은 유사한 형태의 요소이지만 이들의 가장 중요한 차이는 그 가공의 범위가 경계선의 내부인 경우(홀)와 외부인 경우(기둥)의 차이이다. 예를 들어 직경이 10mm 깊이가 10mm인 원형홀의 경우 내경이 10mm, 깊이가 10mm인 원형구멍이 된다. 이렇게 하기 위해서는 직경 10mm 원의 내부를 가공하게 된다. 반면 직경이 10mm 높이(깊이)가 10mm인 원형기둥의 경우 그 외경이 10mm 높이가 10mm가 되도록 가공을 한다는 뜻이다. 이러한 경우 가공은 직경 10mm인 원의 외곽에 행하여진다. 기둥은 모든 가공이 끝난 후 최외곽을 가공하는 경우에 유용하게 쓰이며 홀은 최외곽에 둘러싸인 부분에서의 각종 디자인 요소를 지정하는데 활용할 수 있다. 홀과 기둥은 3가지 유형이 있으며 이들은

1. 원형
2. 다각형
3. 직사각형 들이다.

그림 9는 원형홀 및 원형기둥의 Feature window이다. 이들은 중심, 반경, 깊이 (또는 높이)로 정의할 수 있다. 이와 별도로 6개의 면(TOP, BOTTOM, LEFT, RIGHT, FRONT, REAR) 가운데 하나의 면에 이들이 위치하는 가공면을 지정한다. 홀의 경우 체크박스-관통 이 있다. 이를 체크하는 경우 홀을 가공할 때 최외곽선만 가공한다. 판재를 관통하는 홀을 내는 경우와 같이 내부 가공이 필요 없는 경우 적용할 수 있으며 가공 시간이 줄어드는 장점이 있다.



그림 9. 원형홀 및 원형기둥의 Feature Window

이들 요소들은 추후 사용을 위하여 Library에 저장을 할 수 있으며 (Export) 이름을 지정하고 저장 버튼을 눌러 저장한다. 저장된 요소들은 Import 기능을 사용하여 디자인에 사용할 수 있다.

### 디자인 요소의 변경/복제/삭제/되돌리기

* 디자인 요소의 변경: 기존의 디자인 요소를 변경하고자 하면 해당 요소를 왼쪽 마우스버튼으로 클릭한다. 이 경우 아래의 예와 같이 교체 버튼이 활성화 된다. 필요한 값들을 변경한 후 교체 버튼을 누르면 해당사항이 적용된다.

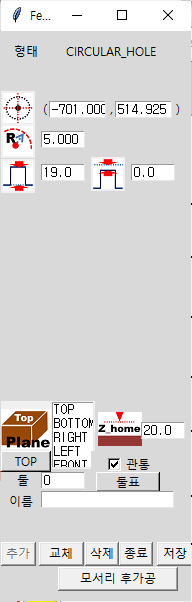


그림 10. 디자인 요소의 변경 시 Feature window의 예

* 디자인 요소의 복제: 디자인 요소를 선택 후 ***ctrl-C*** 를 누르거나  버튼을 눌러 내부에 저장한다. 이후 ***ctrl-V*** 를 누르거나  버튼을 눌러 저장된 요소를 복제한다. 복제된 요소는 원래의 요소에 겹쳐져 보인다. 왼쪽 마우스 버튼을 클릭-> 마우스 이동을 하여 이동을 하거나 Feature window 에서 해당 요소의 각종 값(파라미터)를 변경한 후 ‘교체’ 버튼을 눌러 위치, 크기 등 각종 사항들을 다시 설정할 수 있다.
* 디자인 요소의 삭제: Feature window의 삭제 버튼을 누르거나  버튼 또는 ***Delete Key***를 누른다.
* 되돌리기: 마지막 편집한 내용을 취소하기 위한 것이며 ***ctrl-Z*** 또는  버튼을 누른다.

### 디자인 요소의 대칭/회전

대칭은 디자인 요소를 대칭되는 위치에 새로이 추가하는 기능이다. 이에는 선대칭과 점대칭의 두가지가 있다.

* 선대칭 ( 또는 ‘편집->선대칭’): 디자인 요소를 대칭의 기준이 되는 직선을 기준으로 상대편에 새로이 추가하는 행위이다. 이를 위해서는 디자인 요소와 더불어 기준이 되는 직선이 반드시 있어야 한다.

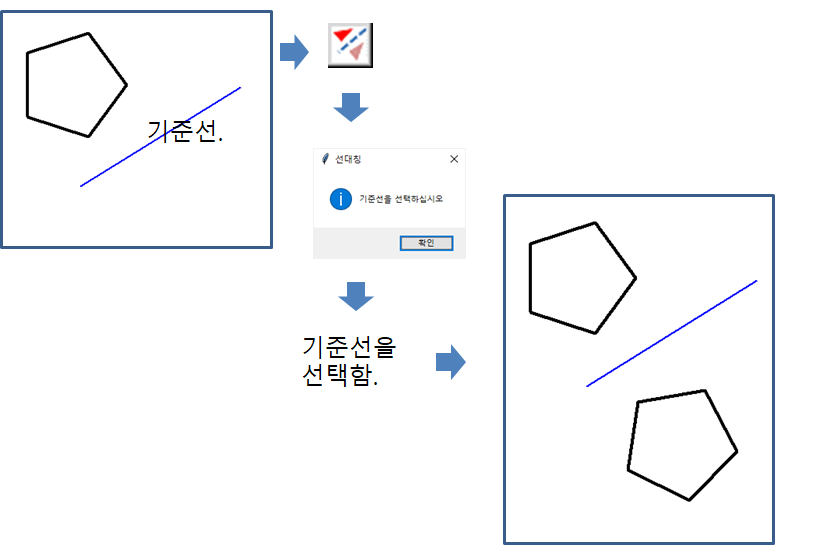


그림 11. 선대칭의 예. 선대칭할 요소를 선택 후  또는 ‘편집->선대칭’으로 시작한다. 기준선을 선택하라는 창을 닫은 후 기준이 될 선을 선택하면 완료된다.

* 점대칭( 또는 ‘편집->점대칭’): 디자인 요소를 대칭의 기준이 되는 점을 기준으로 상대편에 새로이 추가하는 행위이다. 이를 위해서는 디자인 요소와 더불어 기준이 되는 점을 제공할 수 있는 요소가 필요하다. 기준이 되는 요소는 중심을 정의할 수 있는 모든 요소를 사용할 수 있다. 그렇지만 대칭의 기준이 되는 점을 제공하는 요소로는 원 또는 다각형을 사용하는 것을 권장한다.

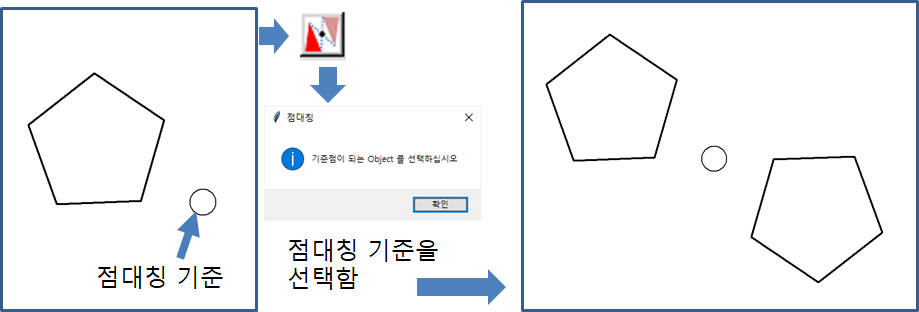


그림 12. 점대칭의 예. 점대칭할 요소를 선택 후  또는 ‘편집->점대칭’으로 시작한다. 점대칭 기준이 될 요소를 선택하라는 창을 닫은 후 원을 선택하여 완료한다. 예에서는 왼쪽의 5 각형홀의 점대칭 요소를 원의 중심을 기준으로 대칭되는 지점에 새로이 추가하였다.

* 회전( 또는 ‘편집->회전’): 디자인 요소를 해당 디자인 요소의 중심을 기준으로 회전시키는 행위이다. 아래의 그림 13은 직사각형 Hole 을 90 도 회전시킨 경우의 예 이다.

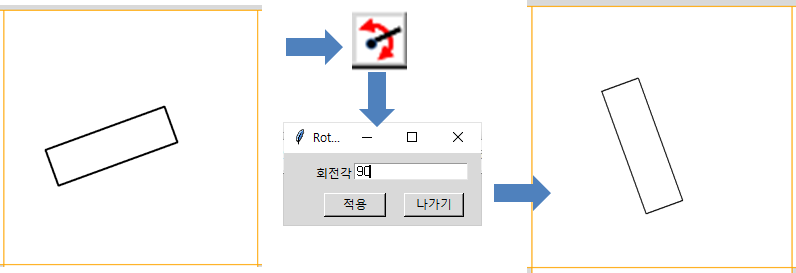


그림 13. 회전의 예. 회전시킬 요소를 선택 후  또는 ‘편집->회전’으로 시작한다. 회전각도를 입력하는 창에서 각도를 “도”의 단위로 입력한 후 적용 버튼을 눌러 완료한다. 예에서는 직사각형의 중심을 기준으로 90도 회전하였다.

### Offset

디자인 요소에 대하여 지정한 offset 만큼 줄어들거나 늘어난 요소를 생성하여 추가한다.  버튼 또는 ‘편집->오프셋’을 통하여 시작할 수 있다. 해당 디자인 요소를 선택하고 offset 버튼을 누르면 아래와 같은 창이 표기된다.

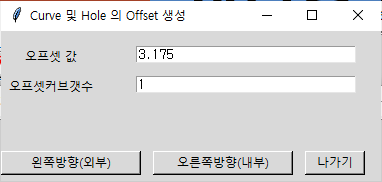


그림 14. Offset 지정 window.

오프셋 값은 축소 또는 확대할 크기를 나타내며 오프셋커브 갯수는 몇 개를 추가할 지 지정하는 Field 이다.

아래의 그림은 디자인 요소에 대하여 offset 을 적용하는 경우의 예를 표기한다.

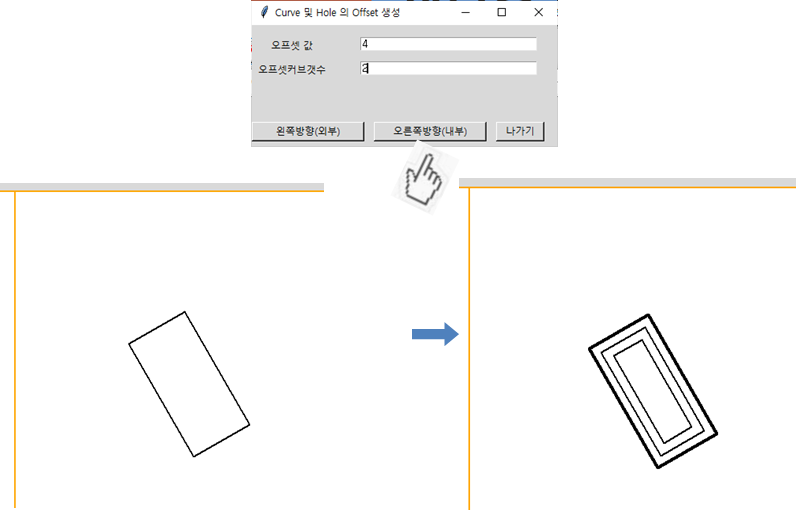


그림 15. Offset 의 예. 왼쪽의 사각형 Hole 에 offset 값 4를 지정하여 두개의 offset 된 요소를 내부에 생성하였다.

# 디자인 요소 살펴보기

디자인 요소는 아래의 버튼 중 모재와 참조선을 제외한 다른 버튼들을 통하여 추가할 수 있다. 모재는 가공을 위한 소재를 지정하는 것이며 참조선은 요소의 크기 상대적 위치 등을 대략적으로 판단하기 위한 눈금을 뜻한다. 따라서 실제 가공에 반영되지 않는다.

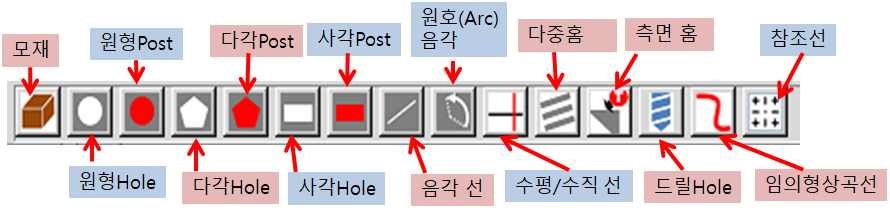


그림 16. 디자인 요소 버튼 및 이들의 기능.

## 원형홀(Circular Hole)/원형기둥(Circular Post)

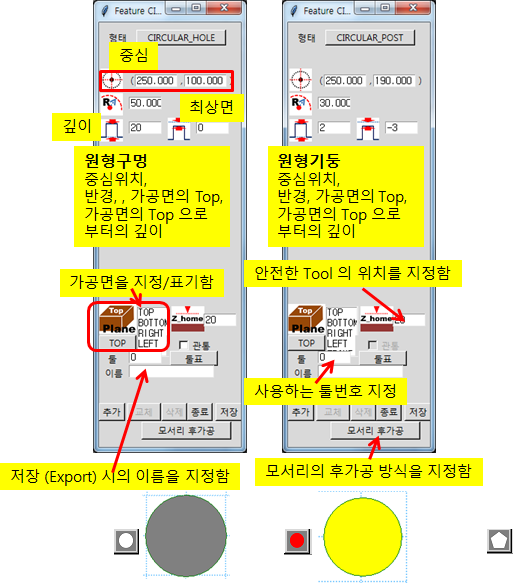


그림 17. 원형홀(Hole) 및 원형기둥(Post)

원형홀 및 원형기둥은 중심, 반경, 깊이(높이), 최상면, 가공면을 통하여 위치 및 크기를 지정한다.

원형홀은  또는 ‘삽입->원형홀’로, 원형기둥은  또는 ‘삽입->원형포스트’로 삽입한다.

* 최상면 : 가공을 할 때 해당 요소의 최상단이 가공 기준면(Z=0 위치)으로부터 어느 위치에 있는지 표기하는 것으로 오른쪽의 원형기둥에서 이를 -3 으로 지정한 경우에 원형기둥의 상부를 기준면에서 3mm 아래까지 깍아 내라는 의미이다.
* 깊이(높이): 최상면을 기준으로 디자인 요소의 깊이(또는 높이)가 얼마나 되는지 표기한다. 오른쪽의 원형기둥의 경우 최상면에서 2mm까지 파내려 가라는 뜻이 된다. 이를 그림으로 표기하면 아래의 그림 18과 같이 표현할 수 있다.

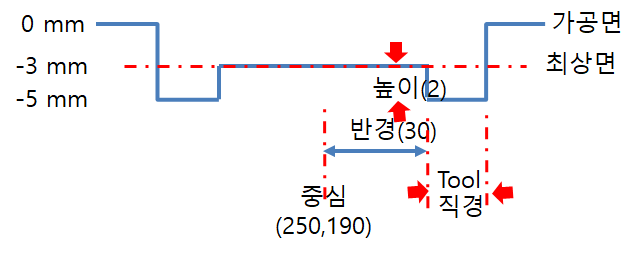


그림 18. 그림 17 오른쪽의 원형기둥 가공시의 각 변수 및 가공 형상. 최상면: -3, 높이: 2, 반경: 30, 중심: (250,190)

* 가공면 : 가공면은 직육면체의 6면중 한 개로 설정한다.
* 툴: 툴표(Tool Table)에 있는 툴의 한 개를 선택할 수 있도록 한다. 툴은 최대 16개까지 저장할 수 있으며 이중 한 개를 택하여 가공에 적용을 하도록 한다. ‘툴표’ 버튼을 누르면 Tool Table이 표기된다. 아래의 그림은 Tool Table이다. 이중 가장 중요한 요소는 직경과 가공량이 된다. 작은 홀을 낼 때 직경이 큰 Tool 을 사용하면 가공이 불가능하므로 디자인 요소의 최소 내경 보다 Tool의 직경이 작아야 함에 주의하여야 한다. 이러한 경우 해당되는 요소는 붉은색으로 표기되어 가공이 불가능함을 알려준다. 가공량은 Tool이 한번 지나가면서 가공하는 깊이를 나타낸다. 예를 들어 깊이가 20mm이고 가공량이 5라고 설정되어 있다면 해당 요소의 가공은 4회에 걸쳐 이루어진다. 이는 Tool의 강도 (재료, 길이 및 직경에 관계), 소재의 강도(단단함의 정도)에 따라 선정을 한다. 단단한 재료를 가공시 이를 크게 한다면 가공중에 Tool이 부러지는 경우가 흔히 발생하므로 사용하는 재료와 Tool에 따라 주의하여 설정한다. Tool table은 주메뉴의  버튼을 통하면 표기와 더불어 그 내용에 대한 변경까지 할 수 있다.
* Z\_home: Tool의 이동시 소재에 손상을 주지않고 이동할 수 있는 안전한 위치를 나타낸다.
* 관통: 소재를 관통하는 가공을 할 때 최외곽에 해당하는 부분만 한번 가공하면 가공이 끝나게 된다. 이러한 형태의 가공시에 ‘관통’을 첵크하면 최외곽만 가공하여 가공 시간을 절약할 수 있다.
* 이름: 설정한 디자인 요소를 저장(export) 하여 재사용(import) 하고자 하는 경우 이를 library에 저장한다. 이때 해당 요소의 이름을 지정하는 칸이다.
* 모서리 후가공: 모서리를 다른 형태의 Tool로 후가공 할 수 있도록 그 방법을 지시하는 것임. 이는 자동 Tool Changer가 있는 경우에 한정적으로 사용할 수 있는 기능이며 별도의 자료로 취급할 예정이다.



그림 19. Tool Table

## 다각형홀(Polygon Hole)/다각형기둥(Polygon Post)

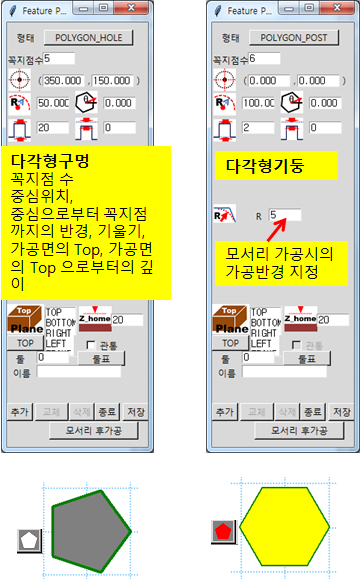


그림 20. 다각형홀 및 기둥

다각형홀은  또는 ‘삽입->다각홀’, 다각형기둥은  또는 ‘삽입->다각포스트’를 통하여 삽입한다. 다각형홀/기둥은 원형홀/기둥과 모든 파라미터를 같이 사용하며 이에 꼭지점의 수 및 모서리가공시의 가공 반경 파라미터가 추가된다. ‘꼭지점수’를 변경하여 3각형 이상의 모든 정다각형을 표기할 수 있다. 반경은 해당하는 정다각형의 외접원의 반경을 의미한다. ‘모서리 가공시의 가공반경’은 모서리를 날카롭지 않게 가공하기 위한 것으로 같은 크기의 다각형홀에 맞는 다각형기둥을 가공하는 경우 홀을 가공하는 Tool의 반경을 입력한다면 이에 꼭 맞는 기둥의 가공을 쉽게 할 수 있게 된다.

## 사각형홀(Rect\_hole)/사각형기둥(Rect\_Post)

사각형홀은  또는 ‘삽입->사각홀’, 사각형기둥은  또는 ‘삽입->사각포스트’로 삽입한다. 사각형은 중심, 수평축 및 수직축의 크기, 수평축의 기울기를 통하여 그 형상을 결정한다. 이에 아래의 그림과 같은 Feature window를 가진다.



그림 21. 사각형홀 및 기둥

## 직선(Line)과 원호(Arc)

직선과 원호는 선형 가공을 하는 요소로 Tool을 움직이며 Tool의 궤적에 따라 모재를 파내는 가공을 하는 요소이다.

* 직선(Line): 일반적인 직선 ( 또는 ‘삽입->선’) 또는 수직/수평선( 또는 ‘삽입->수직/수평선’)을 그릴 때 사용

직선은 시작점과 끝점으로 정의되며 추가적으로 가공 시 Tool의 위치를 지정한다. Tool의 위치는 직선을 기준으로 ‘왼쪽, 오른쪽, 중앙’ 중에서 한 개를 선택하도록 한다. 수직수평선 버튼을 선택한 경우 마우스를 통해 지정한 시작점과 끝점을 잇는 선이 수평에 가까운 경우 수평선으로 수직에 가까운 경우 수직선으로 그려준다.

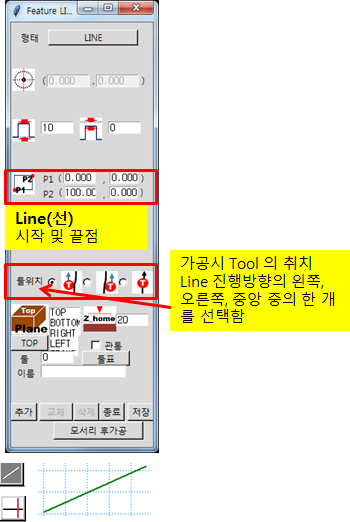


그림 22 직선

* 원호(Arc)

 또는 ‘삽입-> 원호’로 삽입한다. 원호는 중심, 반경, 시작 각도, 원호의 총각도.(끝각도 – 시작각도) 로 표시할 수 있다.

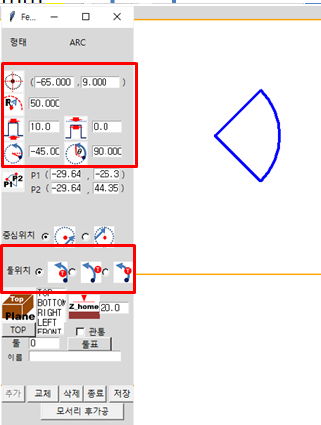


그림 23. 원호 (Arc) : 원호는 중심점, 반경, 시작 각도, 원호의 각도로 정의할 수 있다. 여기에 추가로 가공을 위하여 시작점(P1) 에서 끝점(P2) 으로 이동시 Tool 의 위치를 ‘왼쪽, 오른쪽, 중심’ 중의 한 개로 지정하도록 한다.

## 다중홈(Multi Trench)

간격이 일정한 깊은홈을 가공면에 만드는데 사용한다.  또는 ‘삽입->상면다중홈’으로 삽입할 수 있으며 홈의 넓이는 홈간 간격의 1/2 로 설정된다.

다중홈은 시작점의 위치, 끝점의 위치, 시작 부위에서의 깊이, 끝나는 부위에서의 깊이, 홈의 개수 홈 사이의 간격을 지정함으로써 정의할 수 있다.

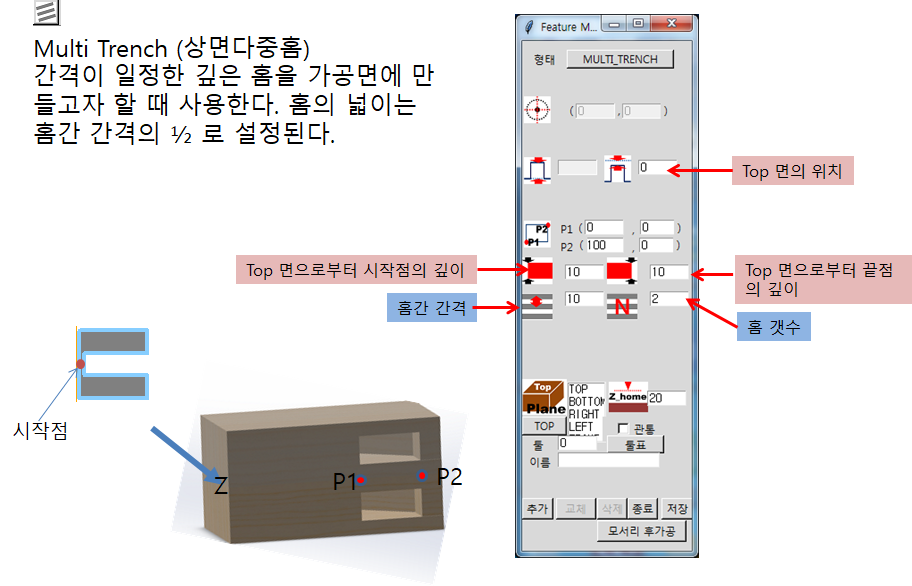


그림 24. 다중홈. 가공면에 간격이 일정한 여러 개의 홈을 만드는데 사용한다.

## 옆면 다중홈(Side Trench) :

 또는 '삽입->옆면다중홈’을 통하여 삽입한다. 직경이 크고 높이가 작은 Tool 을 사용하여 현재 가공면과 수직인 옆면에 상하 간격이 일정한 깊은홈을 만들고자 할 때 사용한다. 홈의 넓이는 홈간 간격의 1/2 로 설정된다

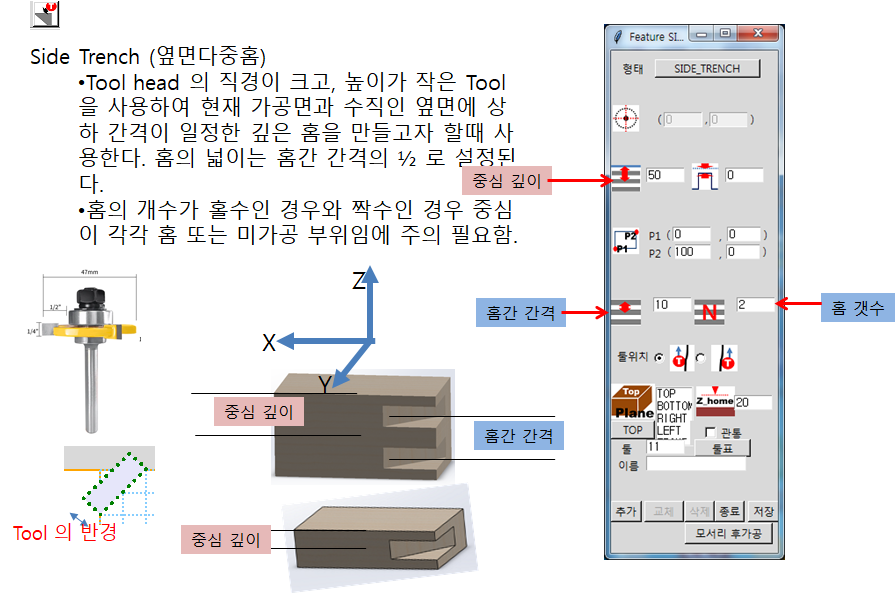


그림 25. 옆면 다중홈. 가공면에 수직인 옆면에 간격이 일정한 여러 개의 홈을 만드는데 사용한다. 이러한 가공에는 옆면을 가공할 수 있는 Tool이 필요하다.

## 드릴홀(Drill Hole)

 또는 ‘삽입->드릴홀’로 삽입한다. 이는 드릴홀을 만들기 위해 사용한다. 이를 통하여 Tool의 직경에 해당하는 홀이 가공된다. 드릴홀은 중심과 깊이 Tool 번호로 정의할 수 있다.

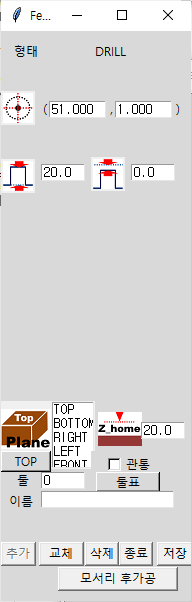


그림 26. 드릴홀

## 임의형상(자유곡선, CURVE)

여러가지 방법으로 곡선을 이루는 Point sequence 를 정의하고 이를 활용하여 임의의 형상을 입력할 수 있는 기능이다.  또는 ‘삽입->임의형상’으로 시작한다. 이 기능은 Woodmeister의 가장 강력한 기능 중의 하나로써 일반적인 CAD에서 제공하지 못하는 임의의 형상을 사용자가 정의하고 구현할 수 있도록 해준다.

Point sequence 는

1. Woodmeister에서 정의한 Script 문법을 이용한 Script file 활용
2. 사용자 프로그램: Python, C 등으로 작성한 사용자 프로그램을 통한 생성
3. Woodmeister 내장 기능의 활용

등 여러가지 방법으로 생성할 수 있다.

이 기능을 시작하면 아래와 같은 Feature window가 생긴다. 모두 5개의 서로 다른 일을 수행할 수 있다. 이중 첫번째 항목인 ‘\*.pts 도형 가져오기’를 사용하여 생성된 Point sequence를 가져올 수 있다.

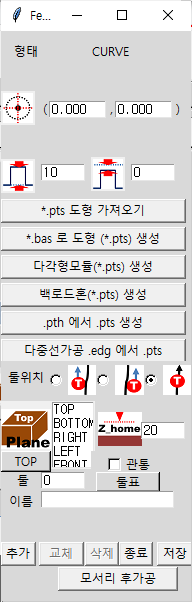


그림 27. 임의형상(Curve)의 Feature window

이의 과정은

1. ‘\*.pts 도형 가져오기’ -> Point sequence 를 가지고 있는 File 선택
2. Offset 입력창(그림 28)에 x, y offset 을 입력한 후 OK
3. Feature window 의 ‘삽입’ 버튼을 누름

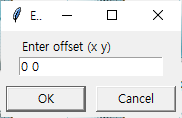


그림 28. Point sequence 삽입 시 Offset 입력 창

아래의 그림은 이와 같은 과정을 거쳐 사전에 생성된 Point sequence를 디자인에 삽입한 예이다.

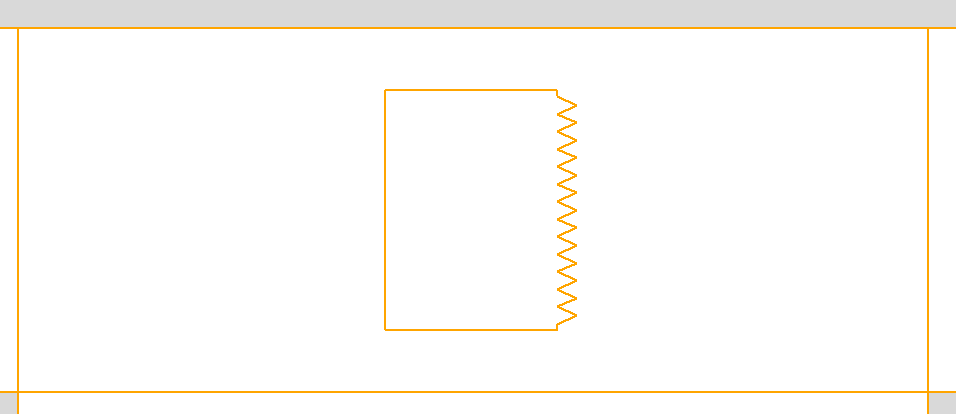


그림 29. Point sequence 삽입의 예

이외의 다른 기능들에 대한 것은 Advanced topic으로 이에 대하여 별도의 자료를 참조하기 바란다.

## 참조선(REF\_LINE)

참조선은 디자인 요소는 아니며 요소들의 대략적인 위치나 크기 등을 파악하기 위하여 사용하는 눈금이다. 이는 Screen Refresh 등을 하면 화면에서 지워진다. 참조선은 중심과 눈금 간격으로 정의할 수 있다.  또는 ‘삽입->참조선’으로 그려 넣는다.

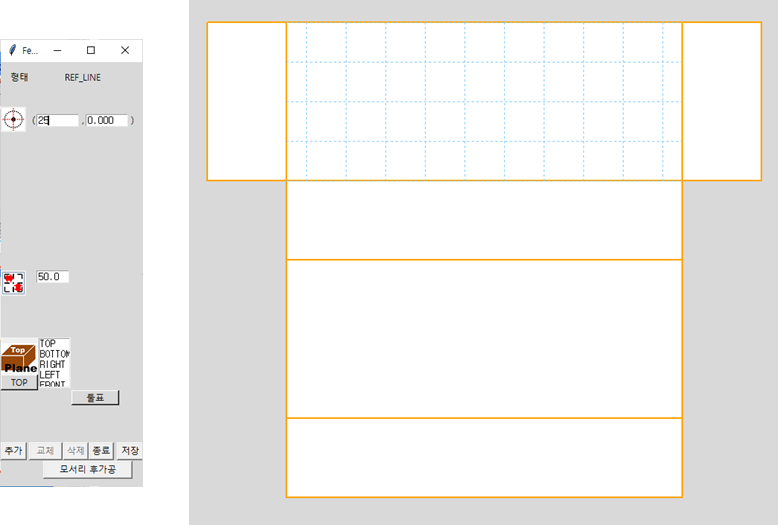


그림 30. 참조선: 참조선은 중심좌표, 눈금간격으로 정의할 수 있으며 6개의 면에 각각 그려넣을 수 있다.

# 라이브러리/문자열/특수도형/사용자 정의 프로그램

## Import

사전에 저장한 가공 데이터, 라이브러리 부품, 디자인 등을 디자인 화면으로 가져오는 기능으로써  또는 ‘화일->가져오기’를 통하여 실행한다.

이를 통하여 .lib, .prt, .json file을 디자인에 추가할 수 있다.

## 특수 도형/디자인 생성

사전에 정의하는 특수한 도형이나 디자인을 생성하는 기능으로써  또는 ‘화일->부품/모듈 생성’을 통하여 실행한다. 일반화 하기 어려운 여러 종류의 디자인 요소 또는 디자인을 생성하는 프로그램을 통하여 구현되었으며 계속적인 변경 및 확장이 있을 수 있어 Advanced Topic 으로 분류된 별도의 자료를 참조하기 바란다.

## 문자열 생성

내장된 기능을 통하여 영문 및 기호들로 구성된 문자열을 생성하고 이를 Point sequence file 로 저장한다.  또는 ‘화일->문자열 객체 생성’을 통하여 실행할 수 있다. 문자열을 입력하고 문자의 넓이를 입력하면 Point sequence 를 생성하여 .pts file 로 저장하여 준다. 저장된 .pts file 은 4.8 절에서 설명된 방법으로 디자인 화면으로 불러올 수 있다.

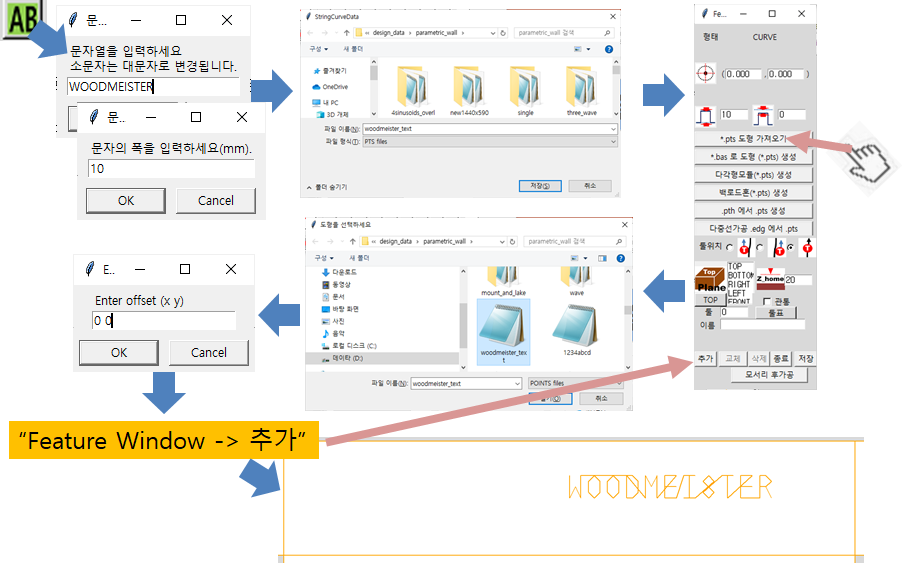


그림 31. 문자열의 생성 및 입력

## 사용자 프로그램 수행

Python을 사용하여 별도로 작성한 사용자 프로그램을 수행할 수 있는 Interface를 제공한다.  또는 ‘화일->Python 프로그램’을 통하여 실행할 수 있다. Woodmeister 프로그램의 변경을 하지 않고 이에 호환되는 각종 data를 생성/저장하거나 활용할 수 있는 통로를 만들기 위한 것으로 일반사용자들의 사용을 권장하지 않는다..

# G-code 생성 및 관련 버튼

 에 포함된 버튼들은 G-code 관련 작업을 지시하기 위하여 사용한다.

## G-code 생성

 또는 ‘G-code-> G-Code 생성/저장’을 통하여 실행할 수 있다. 현재 디자인 window 에 표기된 디자인을 CNC를 구동하기 위한 G-code로 변환하는 작업을 수행한다. 디자인에 사용되는 Tool의 개수와 디자인이 구현된 평면의 개수에 따라 여러 개의 G-code file이 생성된다. 일반적인 1 개의 Tool 및 1 개의 평면(TOP)만을 사용하는 경우가 많아 주로 한 개의 .NC file 이 생긴다.

그렇지만 여러 개의 .NC file을 수용하기 위하여 G-code를 저장하기 위한 Folder 가 생성되며 이 Folder 내에 가공면과 Tool 번호를 포함한 이름을 가지는 여러 개의 .NC file 이 생긴다. 아래와 같은 디자인에 해당하는 G-code를 생성하는 경우 모두 3개의 G-code file 이 생기게 된다.

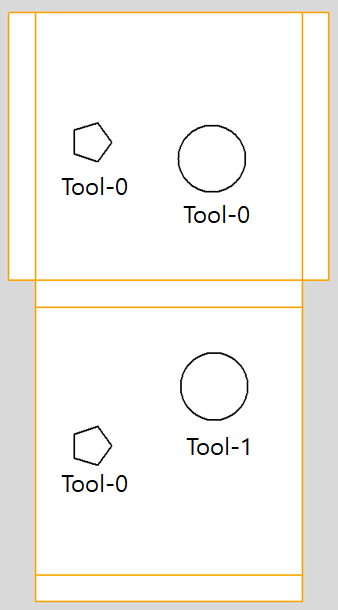


그림 32. 2개의 면 (Top, Bottom) 과 2 개의 tool 을 사용하는 디자인 의 예

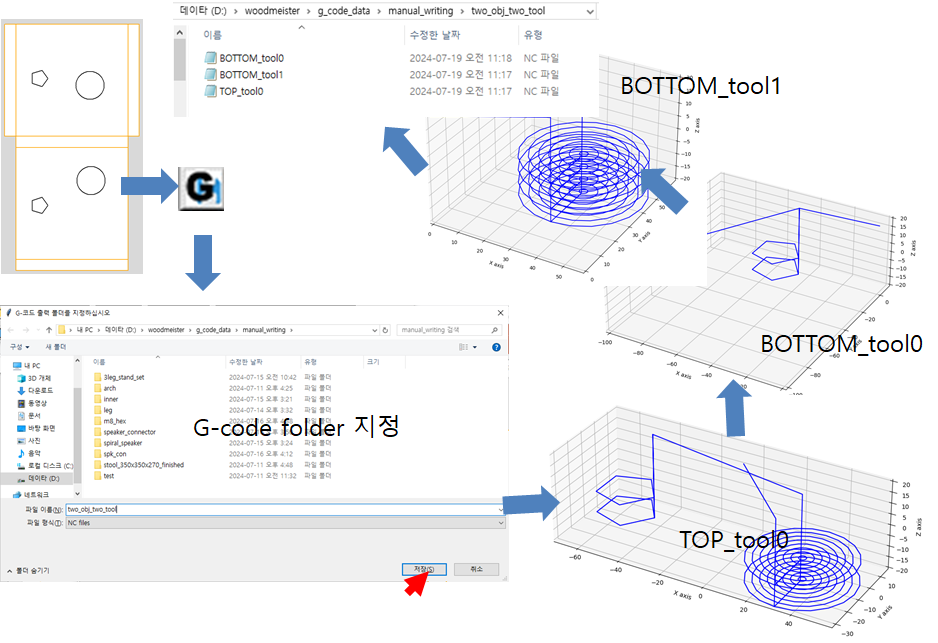


그림 33. 그림 32 의 디자인 으로 부터 생성되어 ../two\_obj\_two\_tool folder에 저장된G-code. 모두 3개의 file 이 생성되었음을 주지하기 바란다.

## G-code 보여주기

생성된 G-code는 디자인을 현실화하기 위한 Tool의 이동 경로를 알려주는 전용의 code 이다. G-code를 시각화 해주는 프로그램을 통하여 이 경로를 볼 수 있으며  또는 Browser 에서 ncviewer.com에 접속하여 해당 G-code가 저장된 File을 열면 그 경로를 보여준다. 그림 34는 two\_surface\_two\_tools/Top\_tool0.NC file 을 보여주고 있다.

## G-code 복제/합성

대형 판재 등 크기가 큰 재료를 사용하여 여러 종류의 디자인 또는 동일한 디자인을 여러 개 가공하고자 하는 경우 사용하는 기능이다.

 를 통하여 시작할 수 있는 기능이며 Advanced Topic 으로써 별도의 자료를 참조하기 바란다.

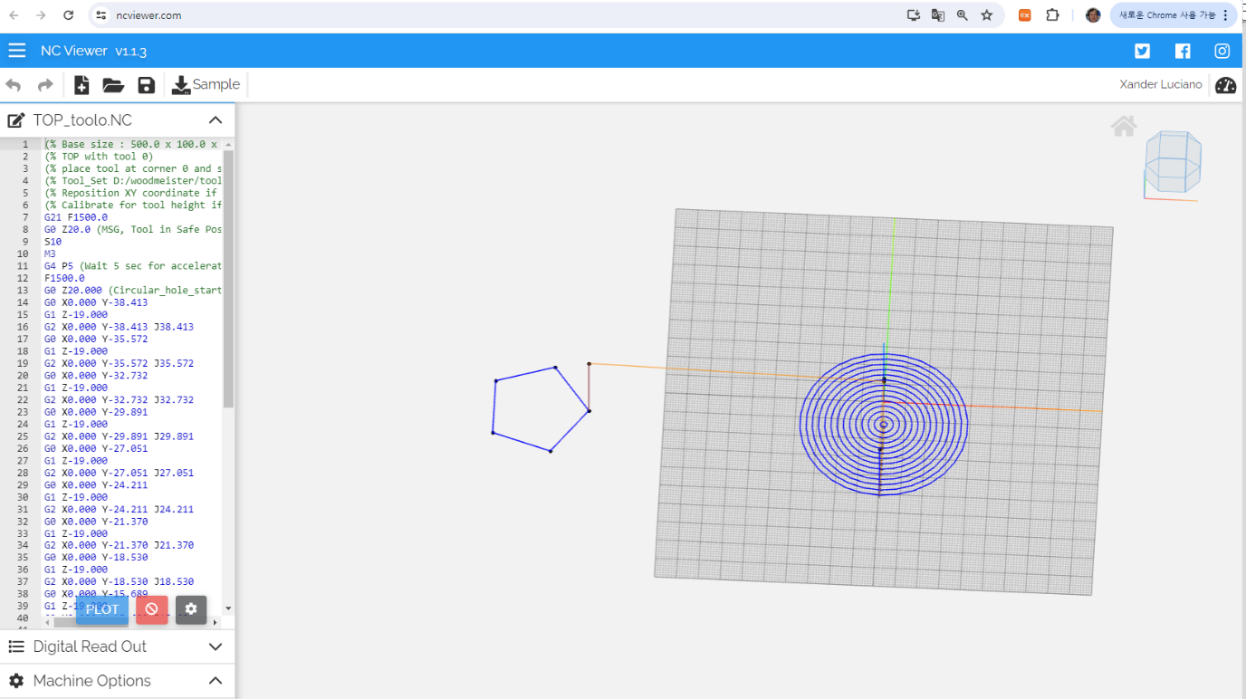


그림 34. On-line G-code viewer : ncviewer.com 에 접속 tool 의 경로를 보여주는 예

## 문자열 생성 – StickFont 사용

Stickfont 를 사용하여 입력 문자를 가공하는 G-code를 생성하며 생성된 G-code는 import를 통하여 디자인에 적용할 수 있다.  또는 ‘화일->문자열 객체 생성’을 통해 시작한다.

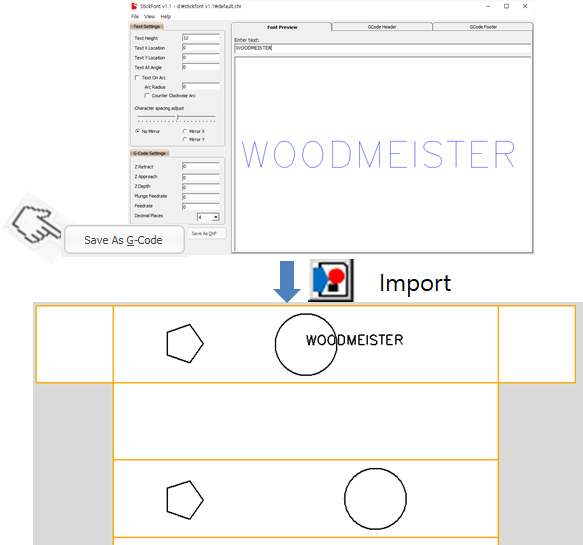


그림 35. Stickfont를 이용한 문자열 발생 및 디자인 적용의 예

# 측정 및 접선

2 개의 디자인 요소에 기반한 디자인 작업을 시작할 수 있도록 한다. 이는  를 통하여 시작한다. 이를 수행하면 두개의 요소를 선택하여 추가 작업을 할 수 있는 창이 뜬다.

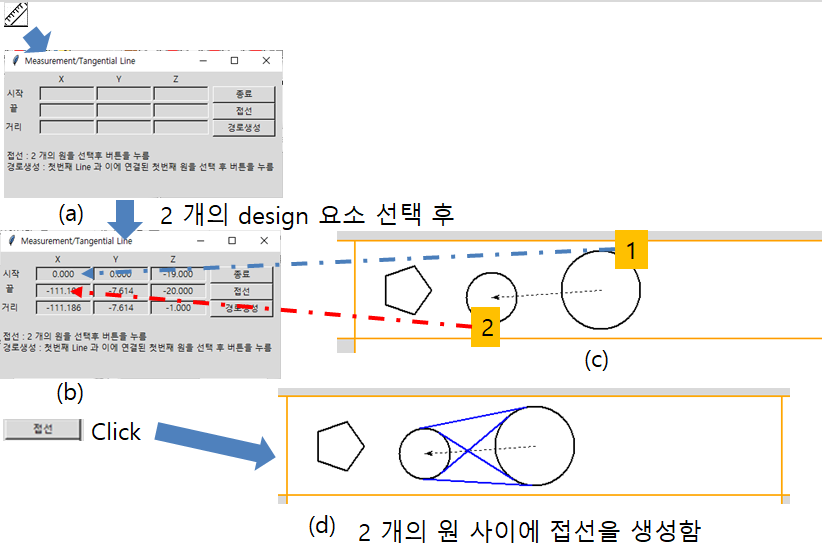


그림 35. ‘측정 및 접선’ 기능:

시작이 되면 그림 35.a와 같은 창이 열린다. 왼쪽 마우스 버튼으로 첫번째 요소를 선택하고 이후 두번째의 요소를 선택하면 그림 35.b와 같이 첫번째 및 두번째 요소의 중심 좌표 (X,Y,Z)가 표기된다. 여기서 Z는 디자인 요소의 깊이를 나타낸다. 이와 함께 디자인 window에 첫번째 요소의 중심에서 두번째 요소의 중심으로 점선 화살표가 이어진다. (그림 35.c) 이로써 두개의 요소에 대한 선택이 끝났다. 두개의 요소가 모두 원형 Hole(Circular Hole) 인 경우, 이 상태에서 ‘접선’ 버튼을 누르면 두개의 원에 접하는 접선이 추가된다. 이와 같은 접선은 호 (원주의 일부)와 접선으로 이루어지는 경로를 만들어 내는데 사용하며 이는 ‘경로생성’ 버튼을 통하여 시작된다. 이를 위한 조건, 용도 및 이후의 사용방법은 11.5.3. “Point sequence file의 생성-접선 경로의 생성”에서 별도로 설명한다.

# Tool table

Tool Table은 사용하는 다양한 Tool에 대한 정보를 보관하고 G-code를 생성할 때 사용하는 Tool data를 내장하는 표이다. Woodmeister는 한 개의 Table에 16개의 Tool을 저장할 수 있다. 이는 디자인에 포함되는 각 요소의 가공에 최대 16개까지의 서로 다른 Tool을 사용할 수 있다는 의미를 가진다. Tool table은 를 통하여 내용을 볼 수 있으며 편집도 가능하다.



그림 36. Tool Table

## 제한사항

Tool table은 중요한 정보를 가지고 있지만 Tool changer가 없는 CNC 기기에서의 사용은 극히 제한적이다. 새로운 요소의 가공 시 기존 요소의 가공 시와 다른 Tool을 요구한다면 Tool을 교체해 주어야 하지만 이를 수작업으로 매번 작업하는 것은 시간적, 안전성 등 여러가지를 고려 시 불가능 함에 유의하여야 한다.

## Tool Table의 항목

* 직경, 가공량: 일반적인 가공을 하고자 할 때 Tool table에서 중요한 값은 직경과 가공량이다. 직경은 가장 작은 Hole이나 Line의 폭을 정하는 요소가 되며 가공량은 한번에 가공할 수 있는 가공 깊이를 뜻한다. 이는 가공의 속도에 영향을 미치는 요소가 된다. 가령 가공 깊이가 20mm인 요소를 가공량 5mm의 Tool로 가공한다면 4번에 걸쳐 깊이를 달리하며 가공을 하여야 한다.
* 길이: Tool의 유효한 길이를 나타낸다. Tool의 길이가 대부분의 경우 가공할 수 있는 최대 깊이를 정하게 된다.
* 오버랩: Tool의 직경보다 폭이 넓은 요소들의 가공시 여러개의 인접한 Tool 경로를 통하여 가공한다. 이때 각각의 경로를 지날 때 Tool의 궤적이 겹치는 양을 지정할 수 있으며 그 값을 나타낸다.
* 오프셋: Tool의 끝 부분에 가공에 사용되지 않는 부분을 가진 Tool 의 경우 해당 부분의 높이를 표시한다.
* 최소깊이 :유효한 가공을 하기 위하여 재료의 표면에서 아래로 내려가야 하는 경우의 최소 깊이
* 마모: Tool 의 사용에 따른 마모량

단단한 재질에 직경이 가늘고 길이가 긴 Tool을 사용할 시 가공중에 Tool이 부러지는 경우가 발생할 수 있다. 따라서 직경, 가공량 등의 선택에는 시험 데이터에 기반한 신중한 선택이 필요하다.

# Group 작업

요소와 겹치지 않는 위치에서 마우스 왼쪽 버튼을 누른 상태에서 마우스를 이동하여 버튼을 release 하면 첫번째 위치와 마지막 위치를 대각선의 꼭지점으로 하는 직사각형 내부의 물체들이 모두 선택되고 이들 선택된 요소들에 대하여 일괄적으로 이동/복사/삭제/대칭/export 등의 작업을 할 수 있다

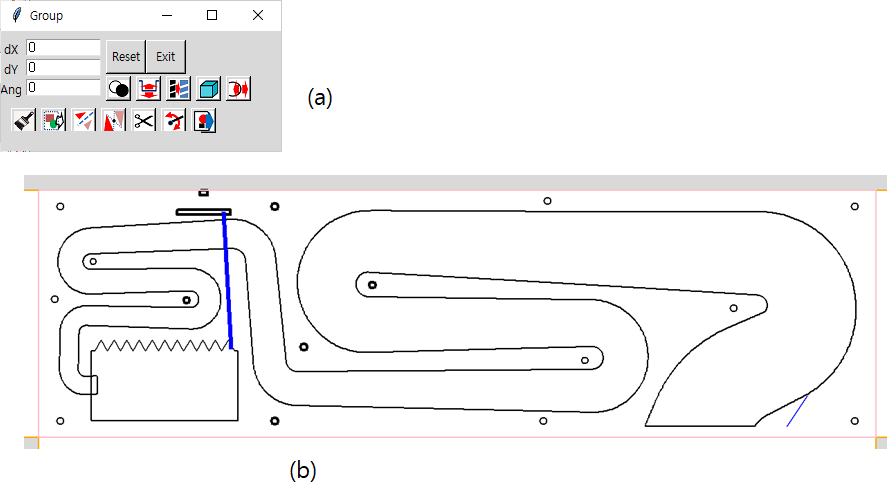


그림 37. Group 선택. 선택된 요소들은 굵은 테두리로 표기된다.

선택된 요소들은 굵은 테두리로 표기되며 이들을  버튼들을 사용하여 일괄적인 처리를 할 수 있다. 그룹에 대한 모든 처리가 끝난 후 Group 창이 남아 있다면 ‘Exit’ 버튼을 눌러 Group 모드에서 빠져나오도록 한다.

## 복사 후 붙여넣기

를 눌러 실행한다. 선택된 모든 요소를 복사 후 붙여 넣는다. 복제된 것들의 원래의 위치에 중첩이 되어 나타나며 이들이 선택된 상태로 된다. 이때 Mouse 왼쪽 버튼을 누른 채 이동하면 복제된 요소들이 이동을 하게 된다. 이동한 양은 dx, dy 오른쪽의 칸에 표기된다. 이 값들을 변경하고  버튼을 누르면 해당하는 값만큼 원래의 위치에서 이동하여 준다. ‘RESET’ 버튼을 누르면 dx, dy 값을 모두 0으로 돌려놓는다.

## 이동

dx, dy 항목에 수치를 입력한 후  를 눌러 실행 하거나. 왼쪽 마우스 버튼을 누른 채 마우스를 이동하여 실행한다. 후자의 경우 이동한 거리가 dx, dy 에 표기된다.

## 선대칭

선택된 모든 요소에 대하여 선대칭된 요소들을 생성/추가 한다. 를 통하여 시작되며 반드시 기준이 되는 직선이 있어야 한다.

## 점대칭

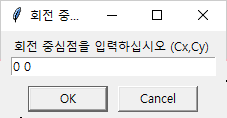
선택된 모든 요소에 대하여 점대칭된 요소들을 생성/추가 한다.를 통하여 시작되며 대칭의 중심이 될 수 있는 요소가 반드시 있어야 한다. 이 중심 요소는 원형의 요소가 가장 바람직하다.

## 요소 삭제

선택된 모든 요소를 삭제한다. 를 통하여 시작한다.

## 요소 회전

선택된 모든 요소를 회전시킨다. ‘Ang’ 항목에 회전 각을 입력한 후 를 누르면 회전의 중심이 되는 좌표를 요구한다.

왼쪽 그림의 예와 같은 방식으로 중심 좌표를 입력한 후 ‘OK’ 버튼을 누르면 선택된 요소들이 회전된 위치로 이동하게 된다. 회전 각도는 도(Degree)로 입력한다. 또한 각도는 반 시계 방향으로 증가한다.

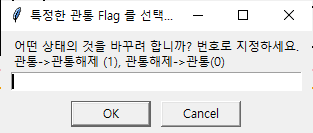
## Export

선택된 요소들을 Library로 저장한다. 저장하는 내용은 파일형식(type)이 .prt 이며 import 를 통하여 사용할 수 있다. 를 통하여 실행한다. 저장시 각각의 요소들은 모두 원래의 위치정보 및 다른 정보를 모두 유지한다.

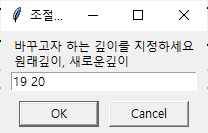
## 조건 부합 요소 변경

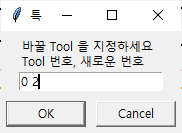
선택된 요소들 중에 조건에 부합되는 요소들 만을 변경하기 위한 버튼 이며 에 속하는 버튼들에 해당한다.

* ‘관통’상태 변경: Feature window 에서 나타나는 ‘관통’ 상태를 변경하기 위한 버튼.

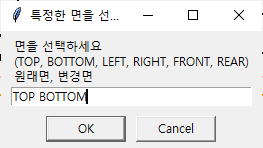
로 실행한다. 왼쪽과 같은 질문 창에서 ‘관통’ 이 check 된 것(True)들을 uncheck(관통 해제)하고 싶으면 ‘1’을 입력하고 ‘‘관통’이 uncheck 된 것(False)들을 check(관통)으로 하고 싶으면 ‘0’을 입력 후 ‘OK’ 버튼을 누른다.

* 깊이(높이) 변경: 특정한 깊이 또는 높이를 가지는 요소들의 깊이(높이)를 다른 값으로 바꾸고자 할 때 사용한다.

로 실행하며 왼쪽과 같은 질문 창에 원래의 값 과 원하는 새로운 값을 입력한 후 ‘OK’ 버튼을 누른다. 왼쪽 창과 같이 입력하였을 때 선택된 요소들 중 깊이(높이)가 19인 것을 찾아내어 깊이(높이)를 20으로 바꾸어 준다.

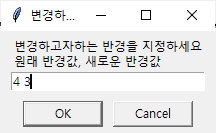
* Tool 변경: 특정한 번호의 Tool이 지정된 요소들을 찾아내어 새로운 Tool을 지정한다.

로 실행하며 선택된 요소 중 특정한 번호의 Tool이 지정된 요소들을 다른 Tool로 교체한다. 왼쪽 그림과 같은 예에서는 tool 0로 지정된 요소들을 모두 찾아 tool 2로 바꾸고자 할 경우의 예이다.

* 면 변경: 특정한 면에 배치된 요소를 다른 면으로 옮기고자 할 때 사용한다.

로 실행하며 선택된 요소 중 지정한 면에 있는 것들을 다른 면으로 옮겨준다. 왼쪽의 예는 TOP에 배치된 것들을 BOTTOM으로 옮겨주게 하는 경우의 입력 예이다.

* 반경 변경: 특정한 값의 반경을 가지는 요소들의 반경을 다른 값으로 바꾸고자 할 때 사용한다.

로 실행하며 선택된 요소 중 특정한 값의 반경을 가지는 요소들의 반경을 새로운 값으로 변경하여 준다. 왼쪽의 예는 반경이 4인 요소들을 골라 이들의 반경을 3으로 바꾸고자 하는 경우의 예이다.

# Keyboard 와 Mouse

Keyboard 기능

|  |  |  |
| --- | --- | --- |
| Key 버튼 | 기능 | 해당 UI 버튼 |
| LEFT | 선택 요소를 좌로 이동 |  |
| Right | 선택 요소를 우로 이동 |  |
| Up | 선택 요소를 위로 이동 |  |
| Down | 선택 요소를 아래로 이동 |  |
| Delete | 선택 요소를 잘라 냄 |  |
| ctrl-C | 선택 요소를 복사함 |  |
| ctrl-V | 선택 요소를 붙여 넣음 |  |
| ctrl-Z | 마지막 Action 을 취소함 |  |
|  |  |  |

Mouse 기능

|  |  |  |
| --- | --- | --- |
| Mouse 버튼 | 기능 | Comment |
| LEFT Click | 요소의 선택 |  |
| Right Press-Drag | 요소의 크기 변경 |  |
| LEFT Press-Drag-Release | 디자인 요소 위에서 시작 시 요소의 이동 |  |
| 디자인 요소가 없는 곳에서 시작 시 group 선택 |  |
| Wheel 위로 굴림 | Zoom In | Zoom은 현재의 Mouse point를 중심으로 이루어짐 |
| Wheel 아래로 굴림 | Zoom Out |
| Wheel press | Zoom을 1:1로 설정 |  |
|  |  |  |
|  |  |  |

# Advanced Topic

Woodmeister는 기존에 많이 사용되고 있는 CAD의 틀을 벗어나 새로운 개념으로 만들어진 CAD 프로그램으로 복잡한 기능은 배제하고 간단히 사용할 수 있도록 구성되어 있다. 이는 목공 등의 간단한 디자인에 쉽게 사용할 수 있는 것을 목표로 하고 있다.

이의 가장 기초가 되는 것은 읽어보고 이해할 수 있는 Text 기반의 구조에 바탕을 두어전용 프로그램을 통하지 않고서도 Notepad 등의 문서편집기로 쉽게 열어볼 수 있고 또한 변경이 가능한 데이터 구조이다.

이 같은 목표를 달성하기 위하여 디자인에 포함되는 각각의 요소(Object)들은 Python Dictionary(Dict)로 표현되며 한 개의 디자인에는 이 같은 요소(Object)들이 최소 1개 이상 포함되어 있다. 자연스럽게 디자인은 이에 포함되는 각 요소들을 표현하는 Dict들의 List로 표현하고 있다. Dict는 ‘:’를 통하여 쌍으로 연결된 여러 개의 Element로 구성되어 있다.

## Object 형식

Woodmeister에서는 여러 개의 Object를 정의하여 사용하고 있으며 dict의 Element 로 여러 종류의 가공 방법 또는 형태를 지시할 수 있다.

### 모재(BASE)

모재는 기본이 되는 소재를 표현한다. 이는 직육면체의 형태를 가정한다. 모든 디자인은 모재가 정의되어 있어야 하며 디자인은 모재를 지정하면서 시작된다. 모재는 디자인내에서 단 한 개만을 지정할 수 있고 디자인의 시작부분에 저장되어 있다. 모재는 가공 요소는 아니다.

{"SHAPE":"BASE", "LONG\_AXIS": 500, "SHORT\_AXIS": 100, "HEIGHT": 100, "Z\_home":20}

"SHAPE":"BASE" -> 모재를 지정함을 나타냄.

"LONG\_AXIS": 500 -> 모재의 X-방향 길이를 나타낸다.

"SHORT\_AXIS": 100 -> 모재의 Y-방향 길이를 나타낸다.

"HEIGHT":100 -> 모재의 Z-방향 길이(높이)를 나타낸다.

“Z\_home":20 -> 이동시 모재에 손상을 주지 않으면서 이동할 수 있는 안전한 높이를 지정한다.

모재와 참조선을 제외한 모든 Object는 가공요소이다.

### 원형홀(Circular Hole)

{"SHAPE":"CIRCULAR\_HOLE", "RADIUS":50, "TOP\_SURFACE": 0, "DEPTH":20, "CENTER":[250,100], "TOOL\_NUMBER":0, "PLANE":"TOP", "Z\_home":20, "PUNCH":False }

이 같은 형태의 데이터는 원형홀을 정의하기 위한 모든 정보를 포함하고 있다.

각 Element는 해당되는 Object를 정의하기 위한 정보들을 담고있다.

“SHAPE":"CIRCULAR\_HOLE” -> 원형홀(Circular Hole)임을 표현한다.

"RADIUS":50 -> 반경이 50mm 임

"TOP\_SURFACE":0 -> 상단면이 0 mm에 위치함.

“DEPTH":20 -> 원형홀을 상단면에서 20mm 깊이로 가공함.

“CENTER”:[250,100] -> 중심좌표를 [250,100]으로 설정함

“TOOL\_NUMBER":0 -> 가공은 Tool-0으로 진행함.

“PLANE":"TOP” -> 6면 중 TOP 면에 위치함.

“Z\_home":20 -> 안전한 Tool의 이동을 위하여 단순한 이동시 Tool은 기준면 (z=0)에서 20mm를 올리고 이동함.

“PUNCH":False -> 'PUNCH’는 ‘관통’을 나타내며 이를 True로 하는 경우 최외곽만 가공하라는 뜻임. 이경우는 내부 가공을 모두 진행함을 지시함.

### 다각형홀(Polygon Hole)

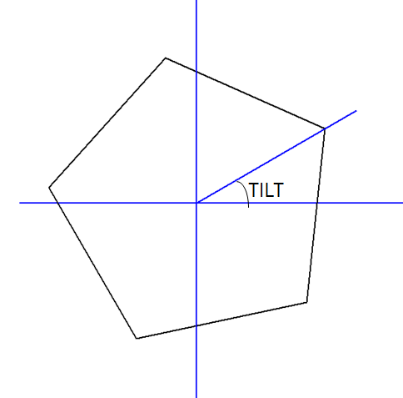
{"SHAPE": "POLYGON\_HOLE", "NUM\_CORNER":5, "RADIUS":50,  
 "TILT":0, "TOP\_SURFACE": 0, "DEPTH":20, "CENTER":[350,150], "CORNER\_CUT":False, "TOOL\_NUMBER":0, "PLANE":"TOP", "Z\_home":20, "PUNCH":True}

다각형홀(Polygon Hole)의 경우 이전의 원형홀과 많은 Element를 같이 사용한다. 다른 Element 형태만 살펴보면..

"SHAPE":"POLYGON\_HOLE” -> 정다각형홀임을 표현함.

"NUM\_CORNER":5 -> 5 개의 꼭지점을 가짐, 즉 정5각형을 나타냄

"TILT":0 -> Tilt는 중심을 지나는 수평선에서 반시계 방향으로 가장 가까이 위치하는 꼭지점과 중심을 잇는 선이 중심을 지나는 수평선과 이루는 각도를 도(Degree)로 나타낸 양이다.

"CORNER\_CUT":False -> 다각형의 내부 가공 시 각 corner는 Tool의 반경으로 인하여 날카로운 corner 형성이 되지 않는다. “CORNER\_CUT”을 True 로 하면 이를 보상하기 위하여 corner의 구석까지 절삭 가공이 이루어지도록 각 corner에서 Tool을 추가 이동하여 가공을 한다.

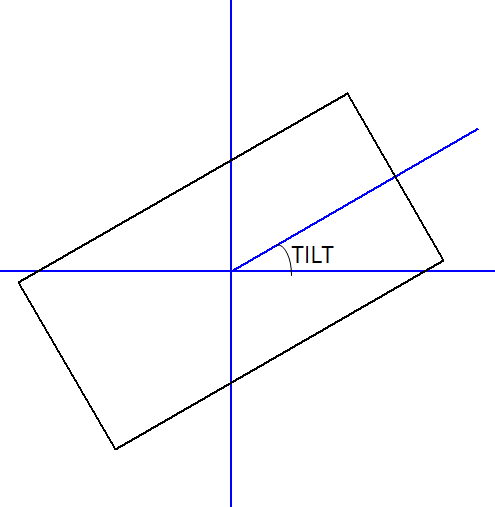
### 사각형홀(Rectangular Hole)

{"SHAPE":"RECT\_HOLE", "LONG\_AXIS":100, "SHORT\_AXIS":50, "TILT":0, "TOP\_SURFACE":0, "DEPTH":20, "CENTER":[350,150], "CORNER\_CUT":False, "TOOL\_NUMBER":0, "PLANE":"TOP", "Z\_home":20, "PUNCH":True}

"SHAPE":"RECT\_HOLE" – 사각형홀임을 표시

새로운 항목들을 살펴보면.

“LONG\_AXIS":100 -> X-방향 축의 길이가 100mm임을 나타냄.

“SHORT\_AXIS":50 -> Y-방향 축의 길이가 50mm임을 나타냄.

“TILT":0 -> 사각형의 경우 Tilt는 LONG\_AXIS 와 수평선이 이루는 각도를 도(Degree)로 나타낸 양이다.

### 원형기둥(Circular Post)

{"SHAPE":"CIRCULAR\_POST", "RADIUS":30, "TOP\_SURFACE": 0,  
"HEIGHT":2, “CENTER":[250,190], "TOOL\_NUMBER":0, "PLANE":"TOP","Z\_home":20 }

"SHAPE":"CIRCULAR\_POST" –> 원형기둥임을 표시

원형홀과 대부분이 유사하며 원형홀의 "DEPTH”가 “HEIGHT"로 대치되었다.

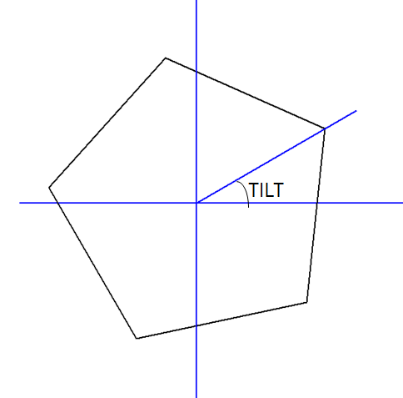
“HEIGHT":2 -> 높이를 2 mm 로 설정함.

### 다각형기둥(Polygon Post)

{"SHAPE":"POLYGON\_POST","NUM\_CORNER":6, "RADIUS":100, "TILT":0, "CORNER\_CUT\_LENGTH":5, "CORNER\_ROUNDING\_RADIUS":5,  
 "CORNER\_RADIUS\_MATING\_HOLE":5, "TOP\_SURFACE": 0, "HEIGHT":2, "CENTER":[0,0],

"TOOL\_NUMBER":0, "PLANE":"TOP","Z\_home":20}

"SHAPE":"POLYGON\_POST” –> 정다각형기둥임을 표현

“CORNER\_ROUNDING\_RADIUS”는 각 꼭지점 부근의 가공 시 지정한 반경을 가진 원호를 이루도록 가공한다는 의미이다. "CORNER\_CUT":False로 표시된 같은 크기의 다각형홀에 맞는 다각형기둥 가공 시 이 값을 다각형홀 가공에 쓰이는 Tool의 반경으로 설정하면 꼭 맞는 다각형기둥을 만들 수 있다.

### 사각기둥(Rectangular Post)

{"SHAPE":"RECT\_POST", "LONG\_AXIS":100, "SHORT\_AXIS":50,  
 "TILT":0,"CORNER\_CUT\_LENGTH":1, "CORNER\_ROUNDING\_RADIUS":1,  
 "CORNER\_RADIUS\_MATING\_HOLE":5, "TOP\_SURFACE": 0, "HEIGHT":2, "CENTER":[0,0], "TOOL\_NUMBER":0, "PLANE":"TOP","Z\_home":20}

“SHAPE":"RECT\_POST” –> 사각기둥임을 표현

### 직선(Line)

{"SHAPE":"LINE", "TOP\_SURFACE": 0, "DEPTH":10, "START":[0,0], "END":[100,0], "POSITION":"RIGHT", "TOOL\_POS\_VEC":[0.7071, 0.7071], "TOOL\_NUMBER":0, "PLANE":"TOP", "Z\_home":20, "CENTER":[0,0]}

“SHAPE":"LINE” –> 직선을 표현

"START", "END” 는 각각 직선의 시작점과 끝점의 좌표를 나타낸다.

“POSITION”은 “RIGHT", "LEFT", "CENTER” 중 하나의 값을 가질 수 있다. 직선의 가공을 위한 이동 시 Tool의 위치를 표기하는 값이다. Tool을 이동 방향의 오른쪽(“RIGHT"), 왼쪽(“LEFT") 또는 중앙(“CENTER")으로 움직이도록 지시하는 Element이다.

"TOOL\_POS\_VEC”은 Tool의 위치 및 직선의 방향에 따라 내부적으로 생성하는 값이며 이부분은 별도의 값을 지정할 필요가 없다.

### 원호(Arc)

{"SHAPE":"ARC", "TOP\_SURFACE": 0, "DEPTH":10, "RADIUS":10, "CENTER":[0,0],"POSITION":"RIGHT", "CENTER\_POSITION":"RIGHT","START\_ANGLE":0, "EXT\_ANGLE":90, "TOOL\_POS\_VEC":[0.7071, 0.7071],  
"P1":[0,45], "P2":[400,240], "TOOL\_NUMBER":0, "PLANE":"TOP","Z\_home":20}

“SHAPE”: “ARC” -> 원호를 표현

“P1", "P2” : “START", "END” 와 같은 의미이며 원호의 시작점과 끝점을 나타낸다.

"CENTER\_POSITION”: 원호를 이루는 원의 중심점이 P1과 P2를 잇는 직선의 오른쪽("RIGHT")인지 왼쪽(“LEFT")인지 지시하는 element 이다.

“START\_ANGLE”: 원호의 중심에서 원호의 시작이 되는 점을 잇는 직선이 수평선과 이루는 각도. 도(Degree)로 표현.

“EXT\_ANGLE”: 원호가 이루는 각도 (STOP\_ANGLE - START\_ANGLE). 도(Degree)로 표현.

### 상면다중홈(MULTI\_TRENCH)

{"SHAPE":"MULTI\_TRENCH", "TOP\_SURFACE": 0, "DEPTH\_START":10, "DEPTH\_END":10, "INTERVAL": 10, "NUM\_COPY":2, "START":[0,0], "END":[100,0], "TOOL\_NUMBER":0, "PLANE":"TOP","Z\_home":20, "PUNCH":False, "CENTER":[0,0]}

"SHAPE":"MULTI-TRENCH"-> 상면다중홈을 표현함.

"DEPTH\_START":10-> 다중홈의 시작 부위의 깊이

"DEPTH\_END":10-> 다중홍의 끝 부위의 깊이

"INTERVAL": 10, -> 다중홈의 중심간 거리 (중심<->중심)

"NUM\_COPY":2, -> 다중홈을 이루는 홈의 개수

### 옆면다중홈(SIDE\_TRENCH)

{"SHAPE":"SIDE\_TRENCH", "TOP\_SURFACE": 0, "DEPTH":50, "INTERVAL":10, "NUM\_COPY":2, "START":[0,0], "END":[100,0], "TOOL\_NUMBER":11, "PLANE":"TOP", "POSITION":"RIGHT","TOOL\_POS\_VEC":[0.7071, 0.7071], "Z\_home":20, "PUNCH":False, "CENTER":[0,0]}

"SHAPE":"SIDE\_TRENCH"-> 옆면다중홈임을 나타냄

“DEPTH":50 -> 옆면다중홈의 수직방향의 중심의 깊이

"INTERVAL":10 -> 옆면 인접한 홈의 중심 간의 거리(수직방향)

"START":[0,0] -> 옆면다중홈의 가공 시 시작점의 좌표

"END":[0,0] -> 옆면다중홈의 가공 시 끝점의 좌표

“POSITION": -> "LEFT”, "RIGHT” 중 선택

“TOOL\_POS\_VEC", "CENTER” 는 입력된 내용에 따라 내부 계산되어 설정됨.

### 드릴홀(Drill-Hole)

{"SHAPE":"DRILL", "TOP\_SURFACE": 0, "DEPTH":20, "CENTER":[0,0], "TOOL\_NUMBER":0, "PLANE":"TOP", "Z\_home":20, "PUNCH":False}

"SHAPE":"DRILL" -> 드릴홀임을 표현함.

### 임의형상(Curve)

{"PLANE":"TOP", "SHAPE":"CURVE", "CENTER":[0,0], "TOP\_SURFACE": 0, "DEPTH":10, "TOOL\_NUMBER":0, "POSITION":"CENTER", "Z\_home":20, "POINTS":[],"MESSAGE":''}

"SHAPE":"CURVE" -> 임의 형상을 표현하는 Point-sequence 를 가지고 있음을 표현

“POSITION": -> point-sequence를 따라 이동시 Tool의 위치를 나타냄. “RIGHT", "LEFT", "CENTER” 중 하나를 선택.

“POINTS": -> 이동할 순서대로 구성된 점들의 좌표를 List로 가짐. 각 좌표는 2차원 (x,y 좌표) 또는 3차원(x,y,action)으로 표현한다. 2차원 좌표로 표현된 경우 절삭가공을 위한 이동을 의미함. 3차원으로 표현된 경우 action의 값에 따라 이동의 방식을 달리함.

Action = 0 -> 주어진 형상의 가공 시 지정된 깊이보다 낮은 깊이로 가공하여 Bridge 를 남겨놓도록 함.

Action = -1 -> 안전한 위치(Z\_home)로 Tool을 이동하여 좌표간 이동을 하도록 함.

### 참조선(Ref-Line)

{"PLANE":"TOP", "SHAPE": "REF\_LINE", "CENTER":[0,0],  
 "INTERVAL": 100}

"SHAPE": "REF\_LINE" -> 참조선을 표현함

“INTERVAL": -> 참조선의 선간 간격

"CENTER"-> 참조선의 중심

## 디자인 File의 구조

디자인 File 의 예

아래의 그림과 같은 디자인을 살펴본다.

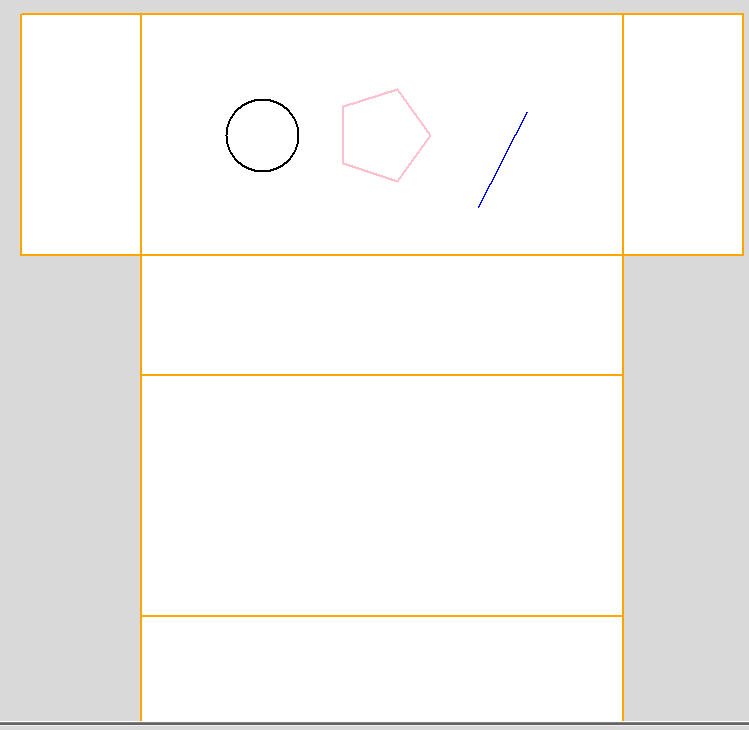


그림 A-1. 참고용 디자인

이 디자인에 해당하는 디자인 file의 내용은 아래와 같이 표현된다.

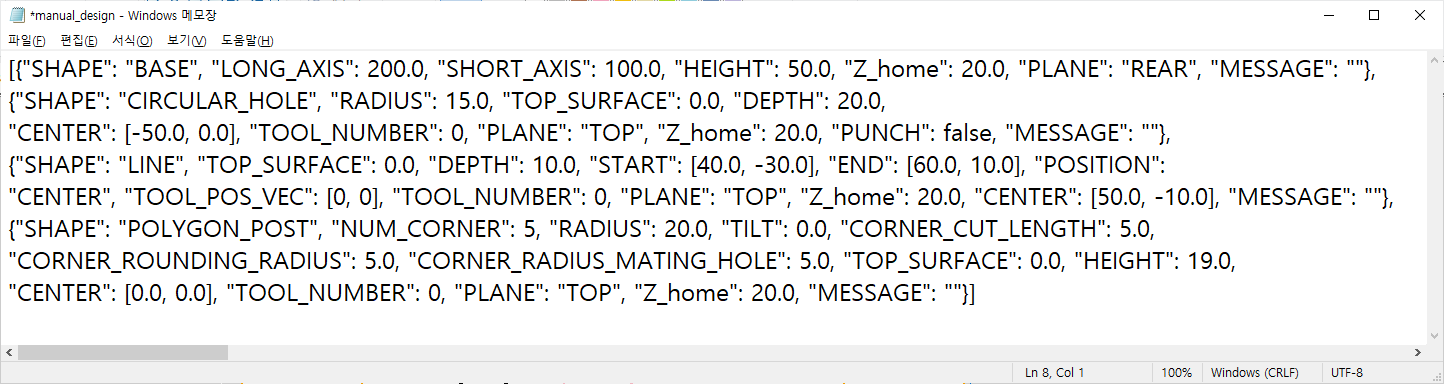


그림 A-2. 참고용 디자인을 저장한 file 의 내용

이 그림을 해석하면 아래와 같이 해석할 수 있다.

{"SHAPE": "BASE", "LONG\_AXIS": 200.0, "SHORT\_AXIS": 100.0, "HEIGHT": 50.0, "Z\_home": 20.0, "PLANE": "REAR", "MESSAGE": ""}, 🡪 모재의 크기는 200x100x50 임

{"SHAPE": "CIRCULAR\_HOLE", "RADIUS": 15.0, "TOP\_SURFACE": 0.0, "DEPTH": 20.0,

"CENTER": [-50.0, 0.0], "TOOL\_NUMBER": 0, "PLANE": "TOP", "Z\_home": 20.0, "PUNCH": false, "MESSAGE": ""}, -> Top 면에 중심:(-50,0) 반경:15mm, 깊이:20mm인 원형홀을 Tool-0로 만든다. 원형홀의 내부 가공을 실행한다.

{"SHAPE": "LINE", "TOP\_SURFACE": 0.0, "DEPTH": 10.0, "START": [40.0, -30.0], "END": [60.0, 10.0], "POSITION": "CENTER", "TOOL\_POS\_VEC": [0, 0], "TOOL\_NUMBER": 0, "PLANE": "TOP", "Z\_home": 20.0, "CENTER": [50.0, -10.0], "MESSAGE": ""}, -> 시작점: (40.0, -30.0), 끝점: (60.0, 10.0)을 연결하는 깊이 10mm의 선을 Tool-0로 가공함. 가공 시 선의 중앙을 따라 Tool을 이동함

{"SHAPE": "POLYGON\_POST", "NUM\_CORNER": 5, "RADIUS": 20.0, "TILT": 0.0, "CORNER\_CUT\_LENGTH": 5.0, "CORNER\_ROUNDING\_RADIUS": 5.0,

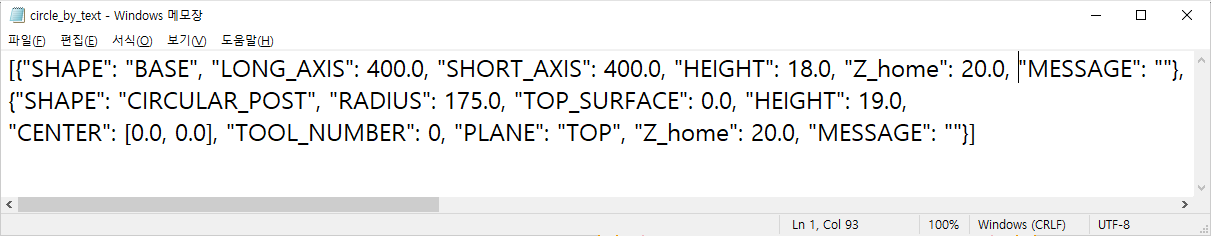
"CORNER\_RADIUS\_MATING\_HOLE": 5.0, "TOP\_SURFACE": 0.0, "HEIGHT": 19.0,

"CENTER": [0.0, 0.0], "TOOL\_NUMBER": 0, "PLANE": "TOP", "Z\_home": 20.0, "MESSAGE": ""} ->외경 40mm인 5각형 post를 높이 19mm로 가공하며 코너를 5mm의 반경을 가지도록 가공함.

이와 같은 예에서 보듯이 디자인 에 포함된 모든 내용은 Text 기반으로 기술되어 있으며 이에 대한 해석이 매우 용이하다.

## 문서편집기를 이용한 디자인 생성

앞에서와 같이 디자인은 Woodmeister를 통하여 작성하는 것이 가장 용이하지만 다른 방법으로도 구현할 수 있다. 간단한 디자인의 경우 노트패드와 같은 문서편집기를 사용하여 충분히 만들어 낼 수 있다. 단적인 예로 400mm x 400mm의 판재를 이용하여 직경 350mm의 원을 만들어내고자 한다면. 아래와 같이 문서를 만들어 저장하면 된다. 만들어진 문서는 circle\_by\_text.json로 저장하였다.



이것은 모재와 원형기둥으로 이루어진 간단한 디자인이다. 이를 Woodmeister를 통하여 열면 아래의 그림과 같은 디자인을 보여준다. 이와 같이 간단한 경우는 문서편집기를 사용하여서도 충분히 디자인할 수 있다.

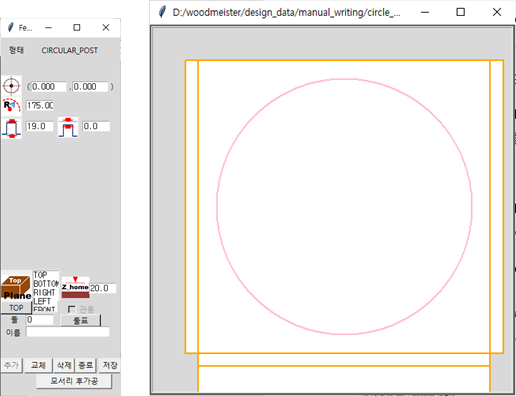


그림 A-3. Circle\_by\_text.json 을 Woodmeister로 열었을때의 화면

다음은 이 원의 중앙에 직경이 40mm인 구멍과 반경 120mm인 원주 위에 4개의 직경 20mm인 원형홀을 뚫기 위한 디자인을 문서편집기로 만들어 본다.

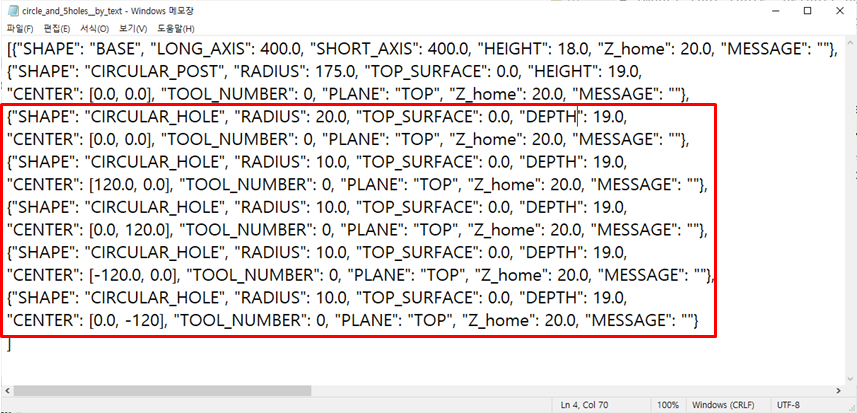


그림 A-4. 원형 Hole 5 개을 추가한 file circle\_and\_5holes\_by\_text.json

5개의 원형홀을 추가한 부분은 붉은색으로 표시한 box 내에 표기되어 있다. 이를 Woodmeister를 통하여 열면 아래와 같은 그림으로 표현이 된다.

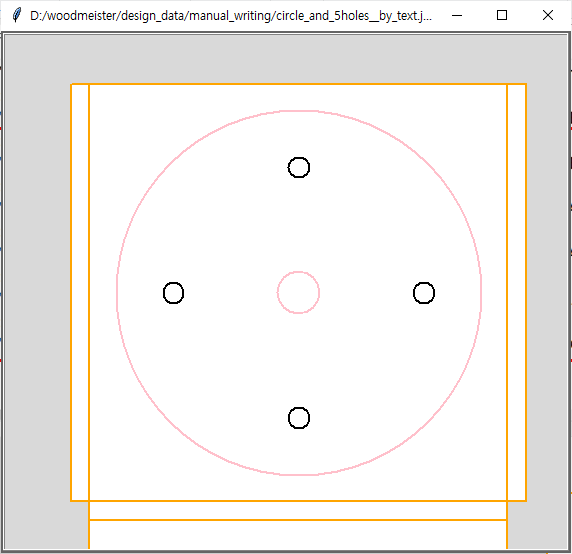


그림 A-5. 그림 A-4를 Woodmeister로 Open한 후의 화면. 디자인에 들어간 요소들이 모두 포현되었음을 알 수 있다.

## Python 프로그램을 이용한 디자인 생성

다음으로 생각하여 볼 수 있는 방법은 프로그램을 통한 디자인의 생성이다. 기본의 구조를 공유하며 이에 적용 상황에 따라 크기, 내부가공물의 위치, 모양 등을 변경하고자 할 경우 충분히 활용할 수 있는 방법이다.

Woodmeister는 python 으로 쓰여진 프로그램이다. 이에 호환되는 data를 python 프로그램을 통하여 생성하는 것이 충분히 가능한 일이다. 이의 기본이 되는 프로그램을 만들어 본다.





그림 A-6. 그림 A.5의 디자인 생성 함수.

이 프로그램은 그림 A-5와 같은 디자인 file을 생성하는 프로그램으로 아래와 같은 구조로 만들어져 있다.

Line 16~20 : 모재를 정의한다.

Line 23~32 : 중앙의 원형홀

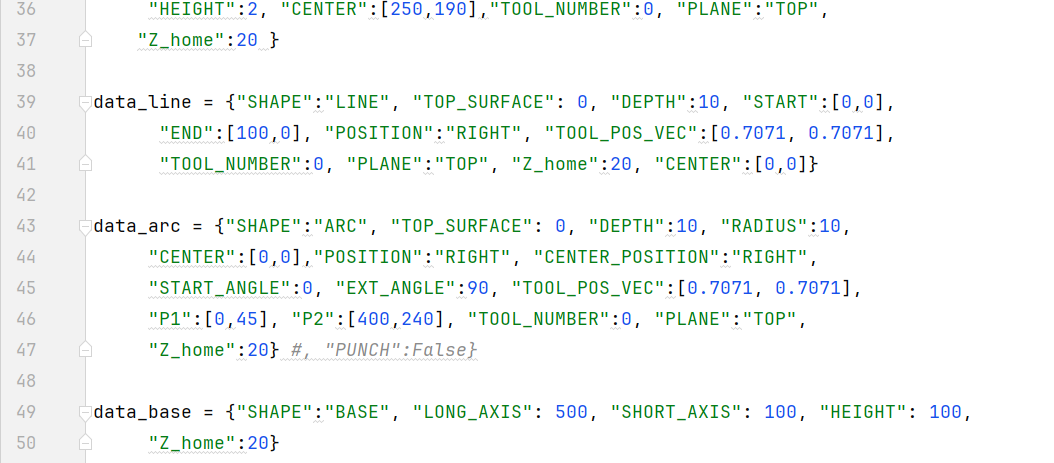
Line 34~ 48 : 주변의 4 개 원형홀

Line51~59 : 외곽을 이루는 직경 350mm의 원(원형기둥)을 정의.

끝으로 data는 사용자의 선택에 의하여 .json format으로 저장한다.

아래의 그림은 geometric\_shapes.py의 내용이다. 이 file은 사용되는 모든 디자인 요소의 Template를 가지고 있으며 이에 기반하여 디자인을 생성한다.

모든 디자인 요소의 template는 다음의 그림과 같다. 이중 일부는 본 설명서에서 다루지는 않으며 이들은 기존에 사용되는 요소로 대체가 가능한 것들이다.



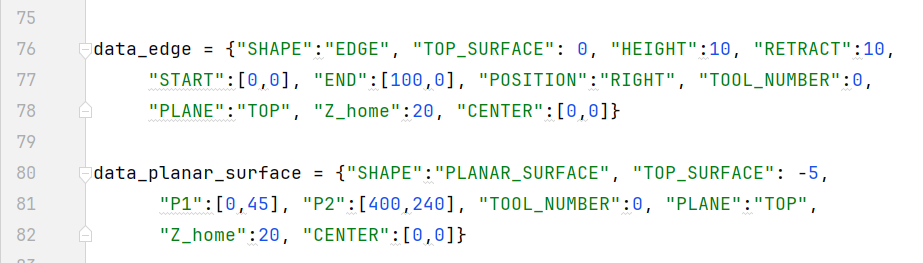


그림 A-7. 디자인 요소의 Template



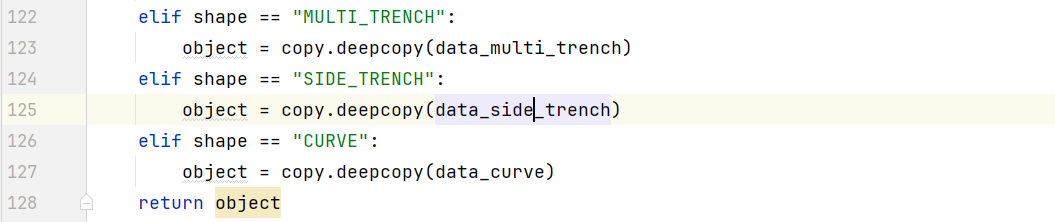


그림 A-8. Template 생성 함수. Shape를 입력하여 해당 디자인 template를 반환 받는 함수. Shape는 각 디자인 요소의 shape 이름이며 반드시 영문으로 하고 “ “ 로 감싸야 한다.

간단한 디자인을 Python 프로그램을 통하여 구현하는 방법을 알아보았다. 이 프로그램을 참조로 하여 좀더 다양한 디자인 요소를 포함하는 디자인을 생성할 수 있을 것이다.

## Point sequence 기반 사용자 정의 자유 형상 구현 –

Woodmeister의 가장 강력한 기능 중의 하나는 사용자가 정의하는 Object를 사용할 수 있다는 것이다. 이는 기존의 CAD에서 거의 제공하지 않는 기능으로 필요로 하는 임의의 형상을 설계하고 가공할 수 있는 길을 열어준다. 이의 바탕이 되는 디자인 요소는 “CURVE” 이다. 앞에서 설명한 디자인 template 중에 임의형상(Curve)에 해당하는 것으로써 이를 이루는 구성요소는

{"PLANE":"TOP", "SHAPE":"CURVE", "CENTER":[0,0], "TOP\_SURFACE": 0, "DEPTH":10, "TOOL\_NUMBER":0, "POSITION":"CENTER", "Z\_home":20, "POINTS":[],"MESSAGE":''}

와 같다. “POINTS”는 임의의 형상을 표현하는 Point sequence이다. 이 Point sequence를 순서대로 이음으로써 임의의 형상을 구현하는 개념이다. 포함된 좌표 값들은 주로 2D로 구성되며 특별한 경우 3D로 표현을 하여 특별한 목표를 지정할 수도 있다.

사용자가 Point sequence를

[[100,100],[100,-100],[-100,100],[-100,100],[100,100]]로 정의하고

이를 크기가 300mmx300mm 두께 18mm인 base 위에 집어넣고 깊이를 19mm로 가정한다면 이를 포함하는 디자인은 아래와 같이 된다.

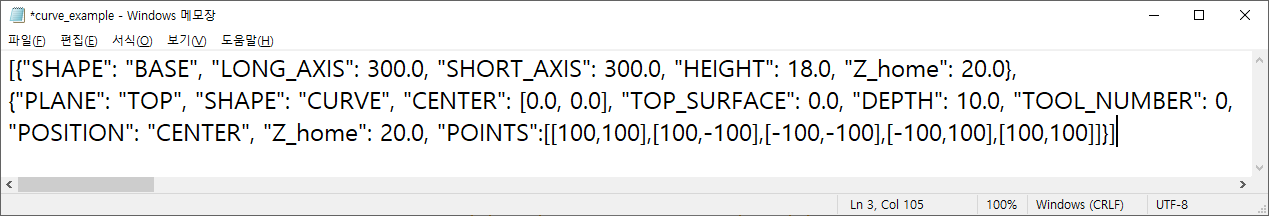


그림 A-9. CURVE를 포함하는 디자인의 예.

Tool-0의 직경이 2mm라 가정한다면 이와 같은 디자인 file로 가공을 진행한 경우 모재의 중앙에 204mm x 204mm의 정사각형 부위를 잘라낼 것이다. 그 이유를 살펴본다면.

Tool이 지나가는 경로를 curve의 중앙으로 하였기 때문에 경로 좌우 측으로 Tool의 반경에 해당하는 2mm씩 잘라낸다. 따라서 모재의 중앙에 204mm x204mm의 사각형홀이 생기고 따낸 사각형은 196mm X 196mm의 크기를 가지게 된다.

이와 같이 디자인 요소로 CURVE를 사용하는 경우 Tool의 직경과 위치를 고려하여 그 경로를 만들어야 한다.

아래의 그림은 그림 A-9를 열었을 때의 디자인 화면이다.

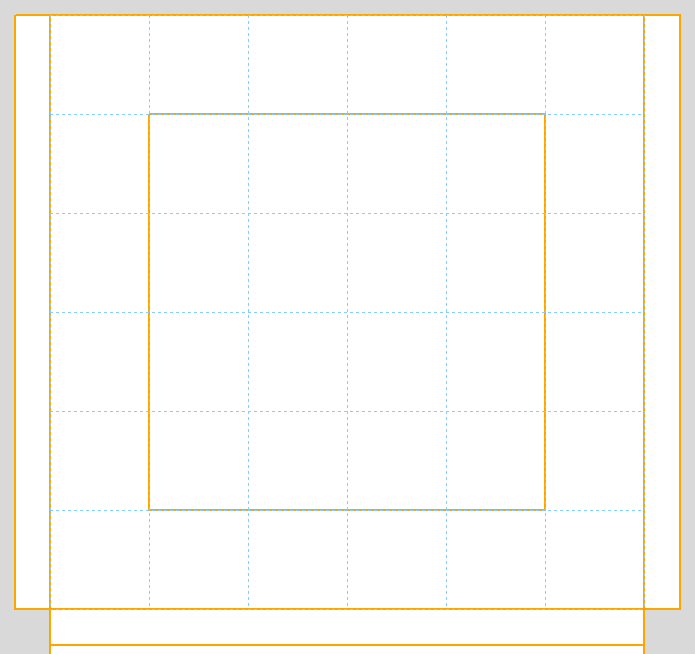


그림 A-10. 그림 A-9의 디자인의 화면. 좌표 값을 알아볼 수 있도록 50mm 간격으로 참조선을 그려넣었다.

### Point sequence 생성 - Script command 활용

사용자 정의 경로를 만들기 위하여 고안한 Script command set로 만든 file (type: .pth 또는 .bld)을 Woodmeister가 읽어 들이고 이를 Point sequence로 변환하는 방법이다.

Command 종류

아래의 예에서 좌표를 명확히 표기하기 위하여 한 점의 x, y 좌표값을 x y 또는 [x, y]와 같이 표현할 수 있다.

* Dimension – 모재의 크기를 지정함  
  DIM W,H -- X-방향 크기 W, Y-방향 크기 H인 모재를 지정함
* line data format  
  LINE sx sy ex ey -- [sx, sy]에서 [ex, ey]를 잇는 선을 만듬  
    
  LINETO ex ey -- 현재의 위치에서 [ex, ey]로 이어지는 선을 만듬.
* bridge data format – Bridge는 가공 시 깊이를 최대로 하지 않고 일부를 남겨 놓음으로써 가공된 요소가 모재에 붙어있는 상태로 하기 위한 것이다.

BRGTO ex ey --현재의 위치에서 [ex, ey]까지 bridge로 만든다.

* bridge에서 남겨놓는 깊이는 common.bridge\_height로 지정할 수 있으며 default 값은 3mm이다. 이 값은 BRGPR 명령으로 바꿀 수 있다. bheight는 남겨놓는 부분의 높이를 나타낸다. 또한 blength는 필요한 경우 bridge의 길이를 지정하는 변수로 사용한다.

BRGPR bheight blength

* 원호: 3 개의 점을 지정하고 이들 점을 지나는 원호의 Point sequence를 생성한다.  
  ARC sx sy mx my ex ey -- 세개의 점 [sx, sy], [mx, my], [ex, ey]를 순서대로 지나는 원호의 Point sequence를 만든다.  
    
  ARCTO mx my ex ey -- 현재의 점에서 시작하여 [mx, my], [ex, ey]를 순서대로 지나는 원호의 Point sequence를 만든다.
* Interpolation 경로 : 몇개의 점을 지나는 Polynomial interpolation 경로를 만든다.  
  INTER [x1, y1] [x2, y2] ... [xn, yn] - [x1, y1] [x2, y2] ... [xn, yn] 의 w점들을 차례로 지나는 interpolation 경로를 만든다.

Interpolation 경로는 포함되는 point의 개수가 많아질 때 경로를 생성할 수 없는 경우가 있다. 이를 해결하기 위하여 point의 개수를 4개 이하로 하는 것이 바람직하다.

* 새로운 경로를 시작함. 기존의 경로에 연결되지 않는 새로운 경로를 시작한다.  
  NEW ex ey -- [ex, ey]에서 새로운 경로를 시작한다.

아래의 그림은 이 같은 Script command 를 가지고 만든 stool.pth file의 예이다.

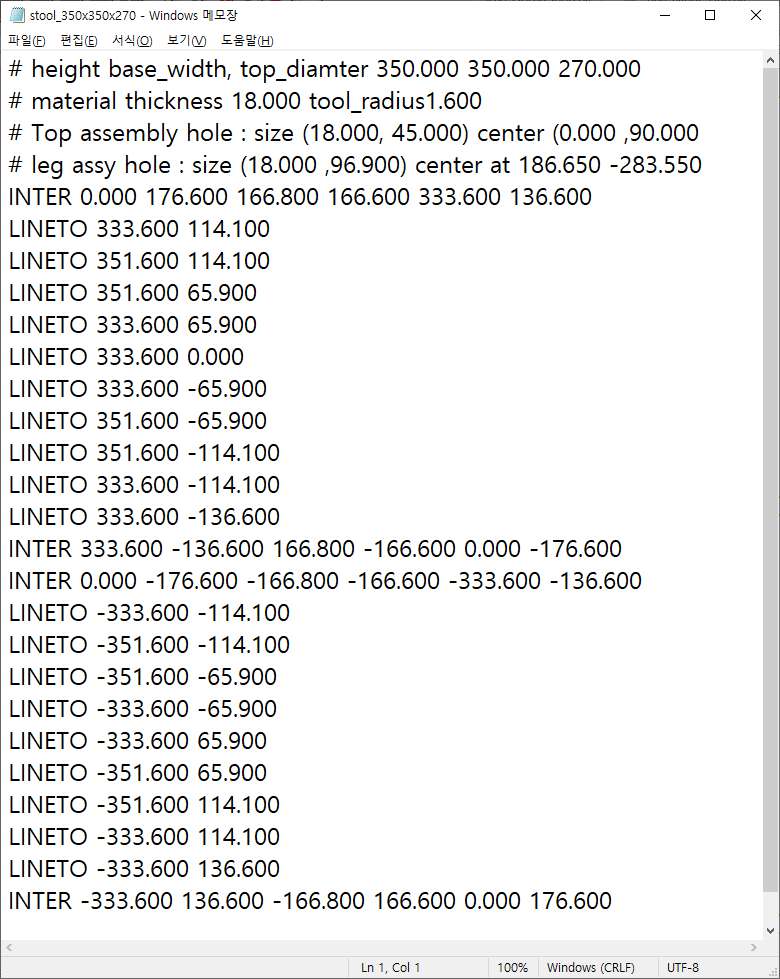


그림 A-11. Script 명령으로 만든 stool\_350x350x270.pth

이를 Woodmeister가 Point sequence로 변환하면 아래의 그림과 같은 형태가 된다.

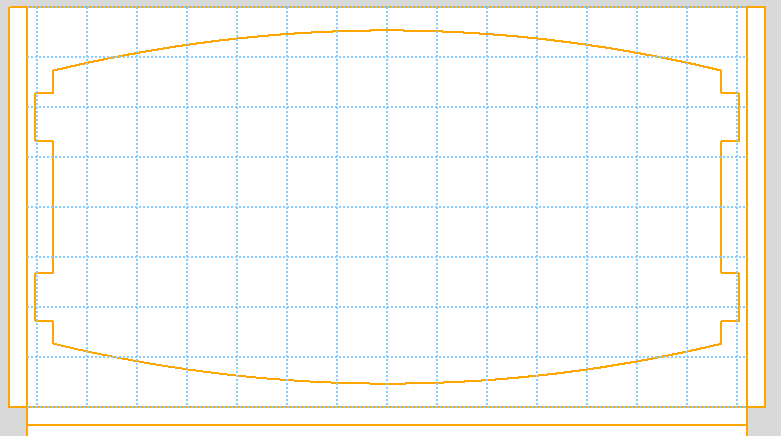
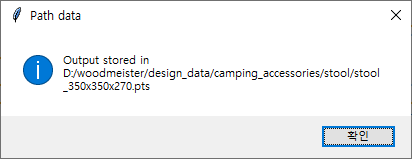


그림 A-12. stool\_350x350x270.pth 에서 생성한 point sequence 로 만든 사용자 정의 요소.

Script command file에서 Point sequence를 생성하기 위해서 다음의 과정을 거친다.

1. 디자인 화면을 열어놓은 상태에서 을 누른다.
2. Pop-up window(Feature window) -> “.pth에서 .pts 생성”을 누른다.
3. Pop-up window(“Select 디자인 file”)에서 stool\_350x350x270.pth를 선택하여 open 한다.

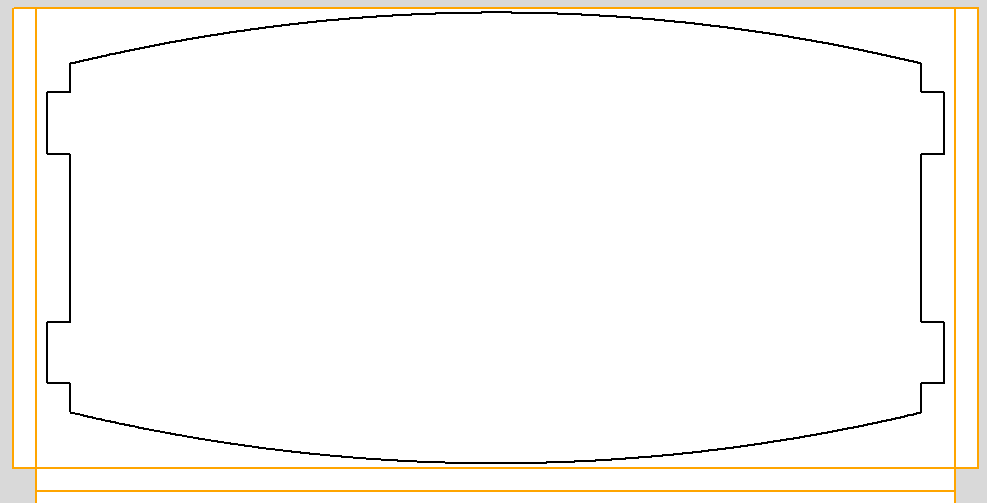


1. 위의 화면에서 확인을 누른 후
2. Feature window -> “.pts 도형 가져오기” 를 누른다.
3. Pop-up window(“도형을 선택하세요” )에서 stool\_350x350x270.pts를 선택한 후 “열기” 를 누른다.
4. Pop-up window(“Enter offset(x,y)”)에서 offset을 입력하고 “OK” 를 누른다..



1. Feature window->“추가”버튼을 누른다.

디자인 window 가 아래와 같이 될 것이다.



1. 삽입이 된 stool\_350x350x270.pts를 click 하면 나타나는 Pop-up window(Feature window)에서 필요한 수치들을 조정하고 “교체” 버튼을 누른다. 아래의 Feature window에서 깊이를 19mm로 조정하였음을 나타낸다.

이와 같이 생성되어 추가한 디자인 요소에 필요한 다른 디자인 요소를 추가하여 디자인을 완성하면 된다. 아래의 그림은 이 같은 결과의 예를 보여준다. 이 예는 휴대용 의자의 디자인 예이다. 이를 이용하여 재단 후 조립을 하면 아래와 같은 형태의 의자를 만들 수 있다.

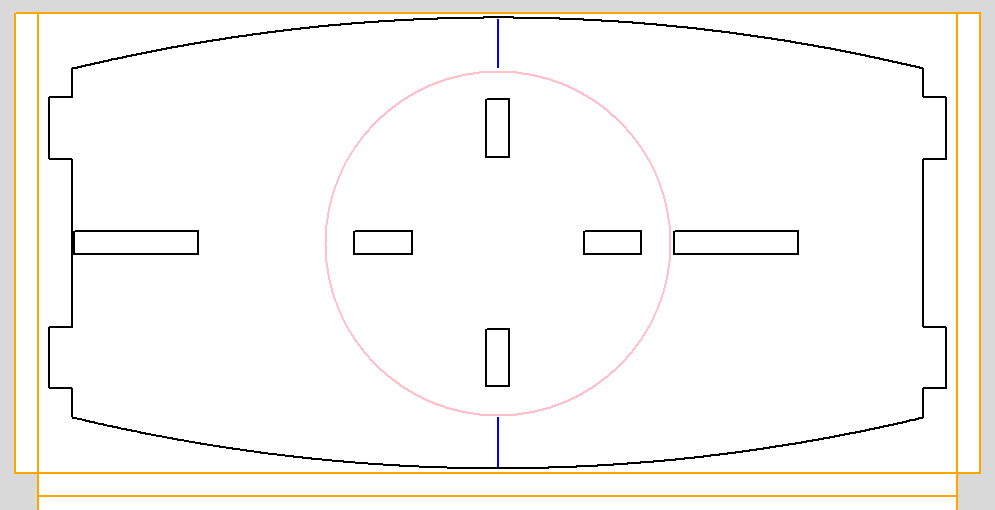


그림 A-13. 사용자 정의 요소를 포함하는 디자인의 예.



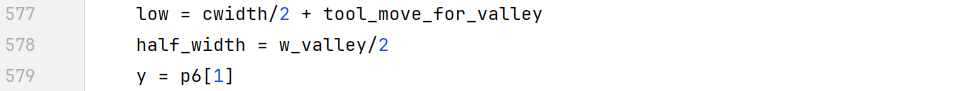
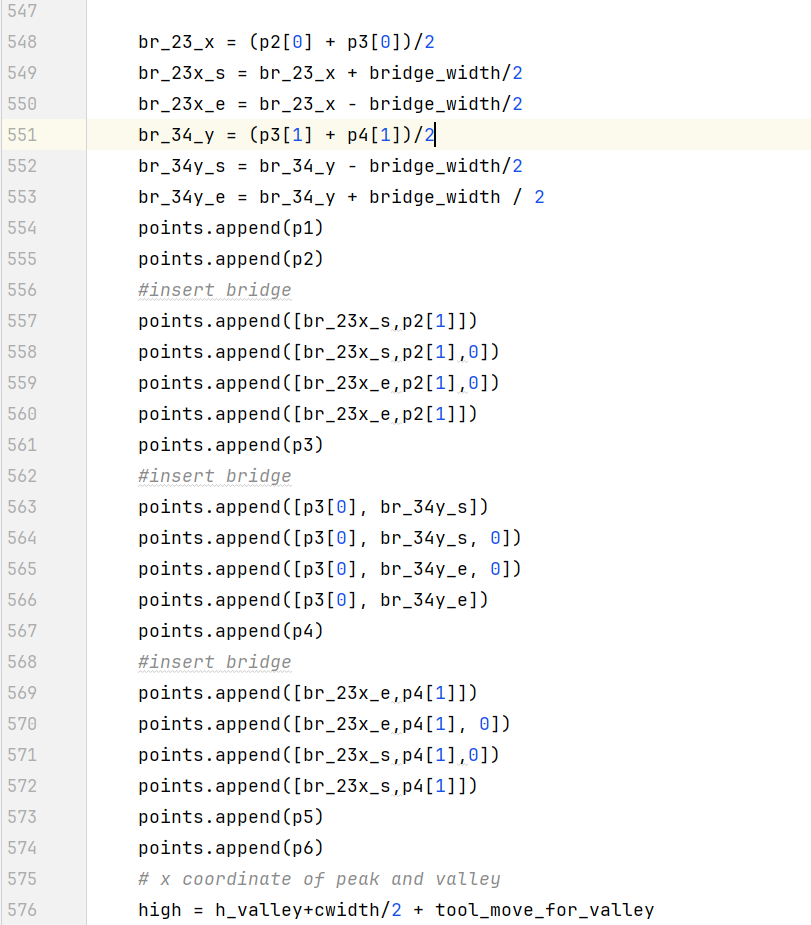
그림 A-14. A-13의 디자인으로 만든 의자

### Point sequence file의 생성 – 직접 편집/프로그램에 의한 생성

사용자 정의 형상 구현은 앞서의 예와 같은 Script command를 이용하는 방법 이외에 직접 .pts file을 문서편집기로 편집하거나 프로그램으로 생성하는 방법도 있다. 이 모든 것을 위하여서는 Tool의 경로 (Tool-path) 및 직경과 같은 내용을 감안하여 적절한 Point sequence를 만들면 된다.

아래의 프로그램은 Backloaded Horn Speaker의 speaker chamber를 생성하는 함수의 예이다. 이는 입력 변수에 의하여 다양한 크기의 chamber를 만드는 프로그램이다.





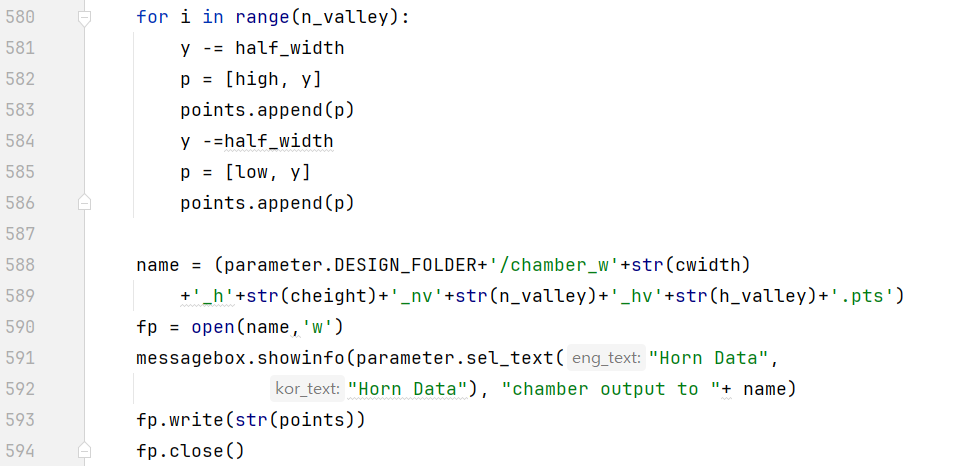


그림 A-15. Speaker chamber 를 생성하는 함수. Point sequence 를 생성하여 .pts file 로 저장함.

아래의 그림은 이 함수를 이용하여 만든 chamber 의 예이다.

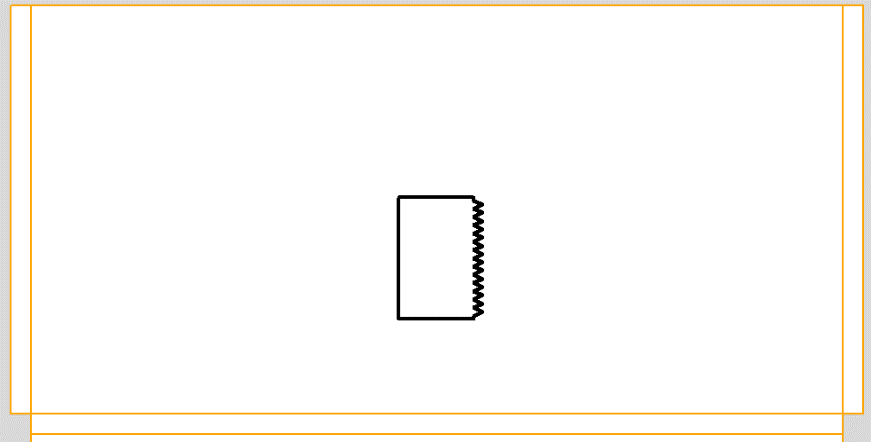


그림 A-16. 생성된 Speaker Chamber 의 예

### Point sequence file의 생성 - 접선 경로의 생성

접선 경로는 원호와 이에 접하는 접선으로 이루어지는 경로로 경로의 방향이 연속적으로 변화하는 경로를 만드는 방법으로 활용할 수 있다. 이의 생성은 몇 개의 단계를 따른다.

그 단계를 순서대로 정리하면

1. 원형홀Circular Hole) 들을 배치
2. 경로를 이루는 순서로 2개의 원형홀 간에 접선을 생성함
3. 각 원형홀 간에 필요한 접선 만을 남기고 나머지 접선들을 삭제함
4. 첫번째 접선과 접선의 끝에 위치한 원형홀을 선택하여 경로를 생성함.
5. 원형홀 배치:

기존의 File을 열거나 새로운 File로 작성함.

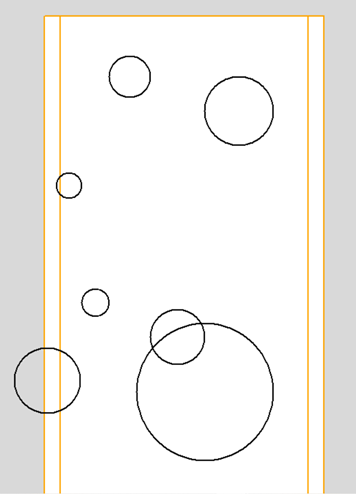


그림 A-17. 원형Hole 의 배치: 기존의 File 또는 새로이 작성함

1. 접선 그리기

를 누른 후 경로가 시작되는 부분부터 순서대로 2개의 원을 선택함.

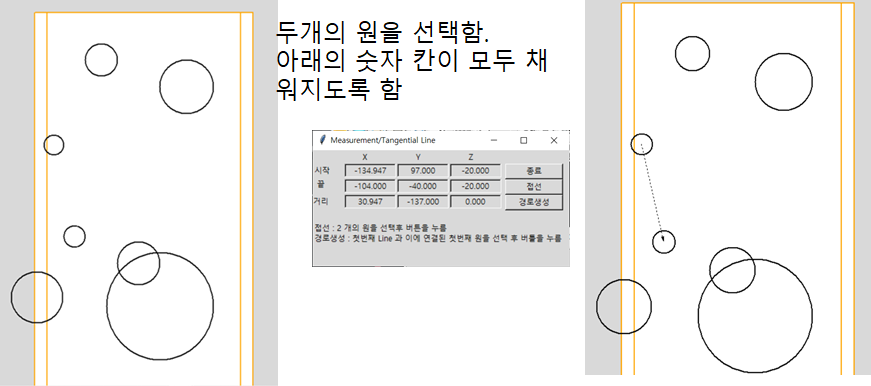


그림 A-18. 접선 그리기. 2개의 원을 선택하면 Pop-up window의 빈칸이 모두 숫자로 채워지며 시작되는 원에서 끝나는 원으로 화살표가 생성됨. 이때 “접선”버튼을 누르면 접선이 생성됨.

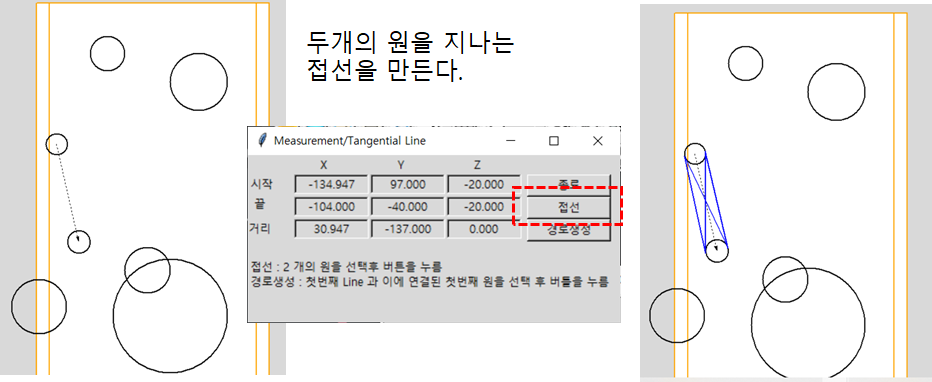


그림 A-19. 두개의 원 사이에 생긴 접선

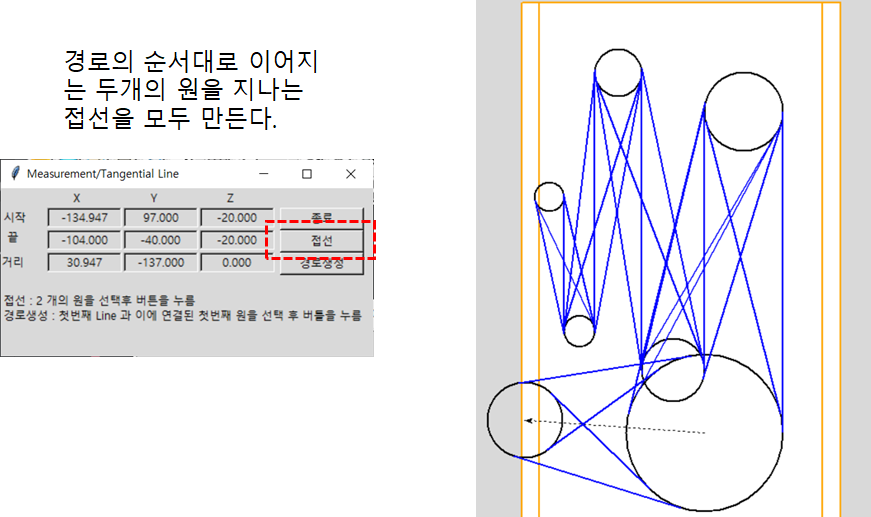


그림 A-20. Pop-up window를 닫지 않고 새로운 원의 pair를 선택하고 이들간의 접선을 경로의 끝에 도달할 때까지 그린다.

1. 필요한 접선만 남기고 나머지 접선을 모두 삭제함.

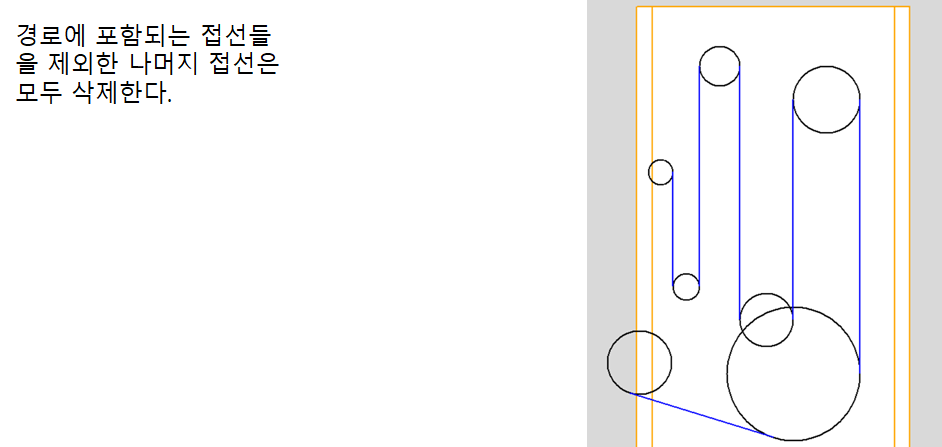


그림 A-21. 필요한 접선만 남기고 나머지 접선을 삭제함.

1. 경로 생성

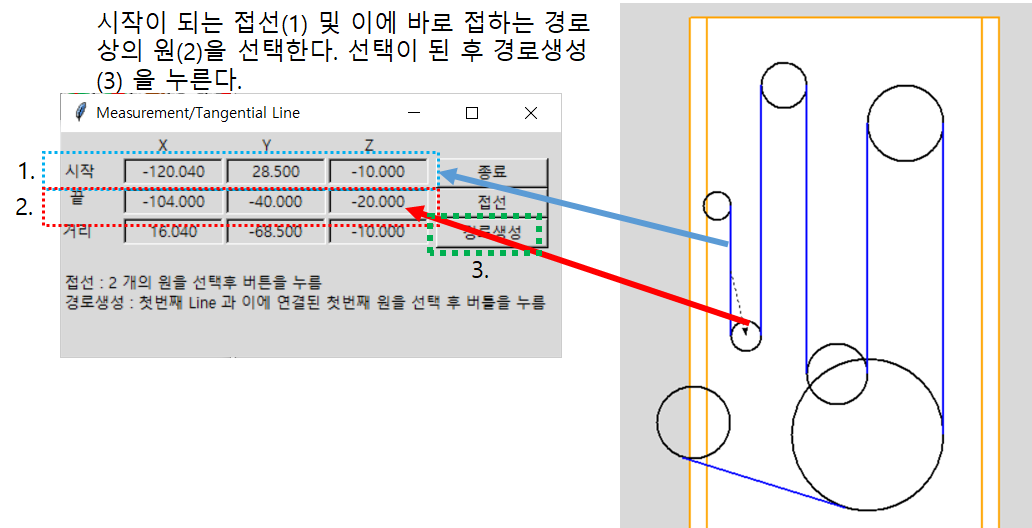


그림 A-22. 경로 생성. 시작이 되는 접선과 경로 상 이의 뒤에 있는 원을 선택한 후 “경로생성”버튼을 누른다.

새로운 Pop-up window가 생기며 가공 시 조각들의 떨어짐을 방지하고자 Bridge를 만들고자 하면 ‘y’ 를 입력한다.

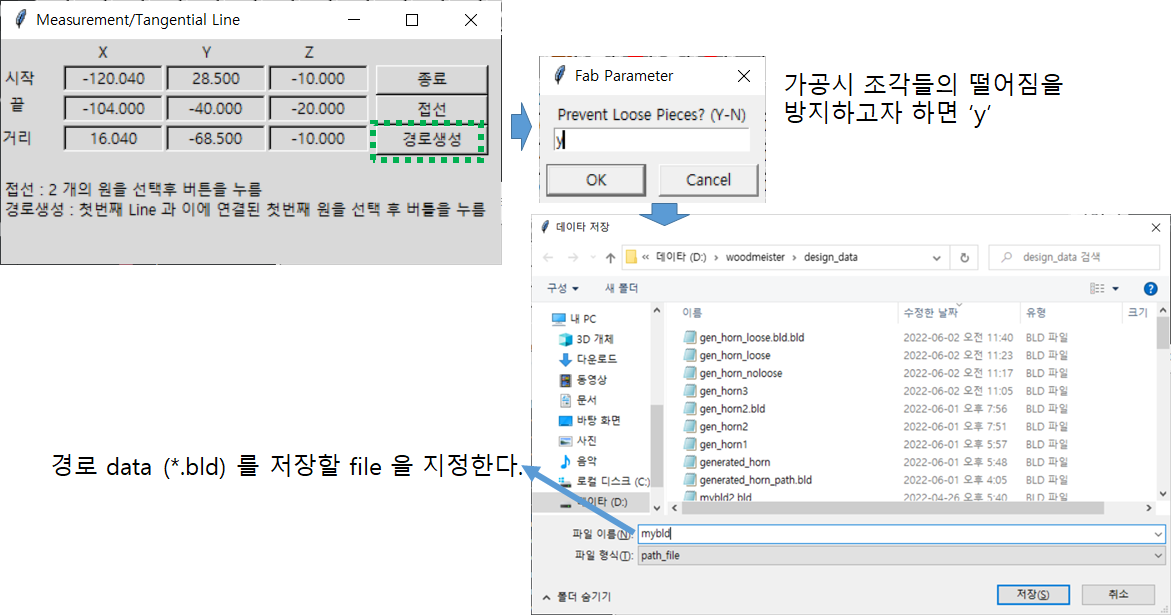


그림 A-23. 경로의 저장. 경로는 .bld file로 저장되며 Script command로 구성되어 있다. 이는 .pth와 동일한 format을 따른다.

보여주고 있는 예에서 저장된 .bld file은 아래와 같다. 필요한 경로를 생성하여 볼 수 있다. 이경우 .bld file은 Backloaded horn speaker의 Horn을 만들기 위한 용도로 사용하기 위하여 별도의 file type을 가지도록 하였다.

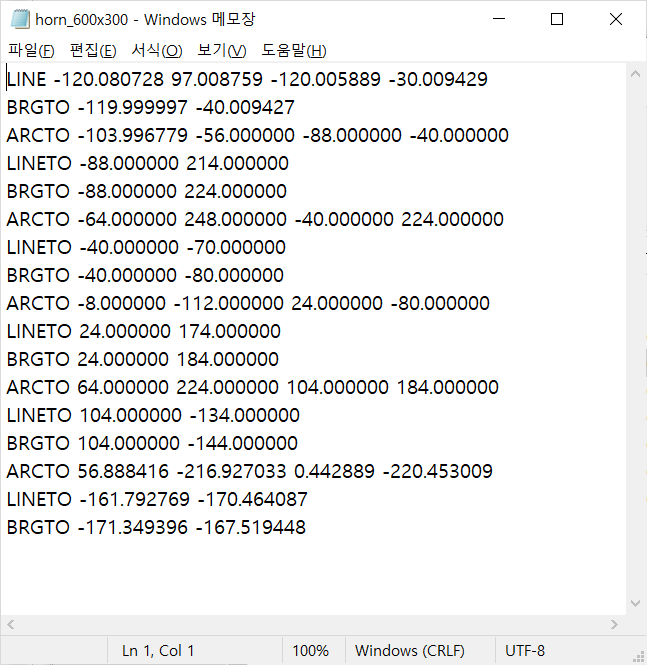


그림 A-24. 경로 File. Horn\_600x300.bld

생성된 경로는 다양한 활용을 할 수 있으며 Backloaded Horn Speaker의 중심 경로로 사용하기에 적합하다. 이 horn\_600x300.bld로부터 생성된 Path는 그림 A-25(a)와 같다.

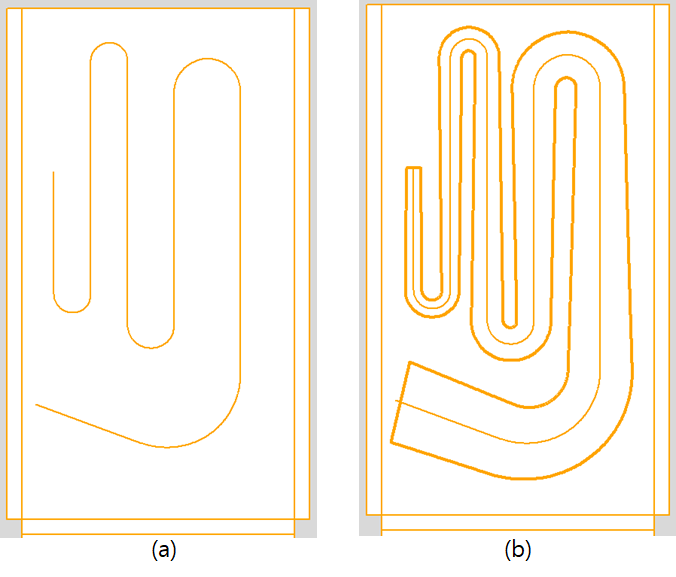
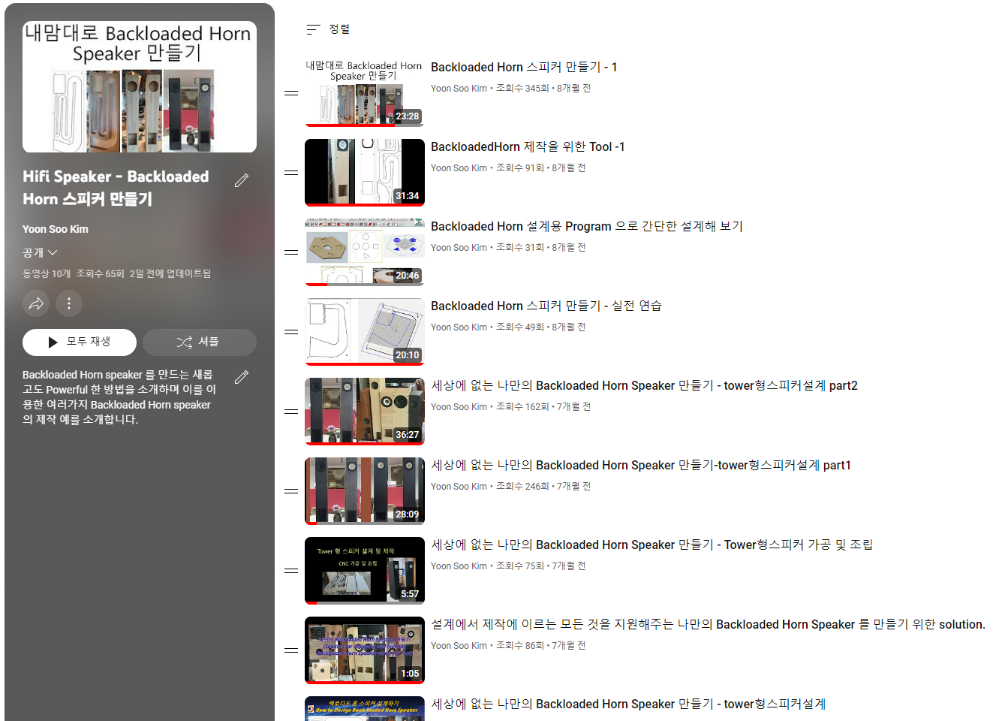


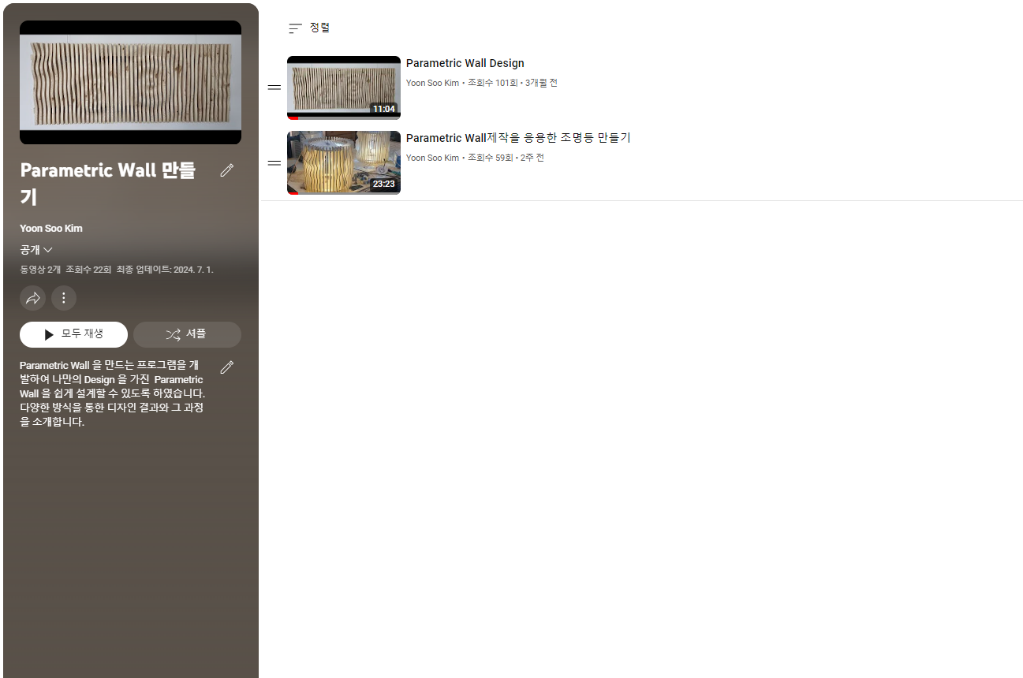
그림 A-25. 그림 A-24의 경로 file로 생성된 경로(a)와 이 경로를 중심 path로 설정하여 생성한 Horn 형상(b). 경로는 원호와 이에 접하는 접선으로 이루어져 있다.

# 부록. Video Contents

본 자료 이해에 참고가 될만한 관련 동영상

[](https://www.youtube.com/playlist?app=desktop&list=PLHzQ8JkLY6_d3xQi7onElSgiXFULc0cMK)

[](https://www.youtube.com/playlist?app=desktop&list=PLHzQ8JkLY6_fipJYY3eOBcDbp1lV-8dA0)

[](https://www.youtube.com/playlist?app=desktop&list=PLHzQ8JkLY6_dPPz5-vaNTMsiyMB2qns64)