

A man in a light blue shirt is seen from the side, holding a tablet. He is in a factory environment with various industrial equipment and a clock in the background. Overlaid on the scene are several digital graphics: a '24/7' icon with a circular arrow, a 'NEWS' section with a person icon, a 'Home' button, and a network diagram with three people icons. The background is a blurred industrial setting with overhead lights and machinery.

SIEMENS

Ingenuity for life

SIMATIC S7-200 **SMART** 直线插补样例— 绘制图像工作流

STEP 7-Micro/WIN SMART V2.7

法律信息

应用实例的使用

应用实例说明了通过文本、图形和/或软件模块形式的几个组件的交互来实现自动化任务的解决方案。本应用程序示例是由西门子公司和/或西门子公司(以下简称“西门子”)的子公司提供的免费服务。它们是非约束性的,并且不声明关于配置和设备的完整性或功能性。应用程序示例仅提供典型任务的帮助;它们并不构成客户特定的解决方案。您有责任按照适用的法规,对产品的正确和安全操作负责,并必须检查相应的应用示例的功能,并为您的系统定制它。

西门子授予您非排他性、不可再授权和不可转让的权利,让经过技术培训的人员使用应用示例。

对应用程序示例的任何更改都由您负责。与第三方共享应用示例,或复制应用示例或摘录,仅允许与您自己的产品结合使用。该应用实例无须接受收费产品的惯常测试和品质检验;它们可能有功能和性能缺陷以及错误。您有责任使用它们,使任何可能发生的故障不会导致财产损失或人身伤害。

免责声明

由于任何法律原因, **Siemens** 不承担任何责任,包括但不限于对应用示例的可用性、可用性、完整性和不存在缺陷以及相关信息、配置和性能数据以及由此造成的任何损害承担责任。这个不适用强制责任的情况下,例如在德国的产品责任法,或意图的情况下,重大过失,或有罪的生命损失,人身伤害或损坏健康,不符合担保,欺骗性的非披露缺陷或有罪的违反合同义务。但因违反重大合同义务而提出的损害赔偿要求应限于协议类型的典型可预见损害,但因故意或重大过失或基于生命损失、身体伤害或健康损害而产生的责任除外。上述规定并不意味着对你不利的举证责任的任何改变。对于第三方在此方面的现有或未来索赔,您应向西门子作出赔偿,除非西门子负有强制责任。

通过使用应用示例,您承认西门子对上述责任条款之外的任何损害不承担责任。

其他信息

西门子保留随时更改应用示例的权利,无需另行通知。如果应用实例中的建议与其他西门子出版物(如目录)之间存在差异,则应优先考虑其他文件的内容。

安全信息

西门子提供具有工业安全功能的产品和解决方案,支持工厂、系统、机器和网络的安全运行。

为了保护工厂、系统、机器和网络免受网络威胁,有必要实施——并持续维护——一个整体的、最先进的工业安全概念。西门子的产品和解决方案构成了这一概念的一个元素。

客户有责任防止对其工厂、系统、机器和网络的未经授权的访问。

这些系统、机器和组件只应在必要的情况下连接到企业网络或 **Internet**,并且只有在适当的安全措施(例如防火墙和/或网络分割)到位的情况下才应连接到这种连接。有关可能实施的工业保安措施的其他资料,请浏览

<https://www.siemens.com/industrialsecurity>.

西门子的产品和解决方案经过不断的发展，使其更加安全。西门子强烈建议，一旦产品更新可用，就立即应用产品更新，并使用最新的产品版本。使用不再受支持的产品版本以及未能应用最新更新可能会增加客户遭受网络威胁的风险。

要了解产品更新，请订阅西门子工业安全 RSS 源，网址为：

<https://www.siemens.com/industrialsecurity>.

目录

1	概述.....	5
2	要求及其他工具.....	6
2.1	PLC 软硬件要求	6
2.2	其他工具	6
3	workflow 概览.....	7
3.1	Vector Graphics.....	7
3.2	G-Code	7
3.3	PathData.....	7
4	获取矢量图文件.....	8
4.1	Adobe Illustrator 简述	8
4.2	操作步骤	9
5	获取 G-Code.....	14
5.1	ABViewer 简述.....	14
5.2	操作步骤	14
6	获取路径坐标点.....	18
6.1	Python 安装环境	18
6.2	Python 代码运行	20
6.2.1	Visual Studio Code 简述.....	20
6.2.2	程序执行	20
6.2.3	执行结果	22
7	简单的坐标获取方法	24
7.1	AutoCAD 简述	24
7.2	操作步骤	24
8	执行结果	28

1 概述

S7-200 SMART V2.7 版本新增加了轴组功能，三轴的运动坐标系支持笛卡尔运动控制模型（ST20 仅支持两轴插补），在软件中拖拽向导生成的子例程即可使用，通过直接给定坐标点和速度参数的方式的实现插补功能。

同时 S7-200 SMART V2.7 版本还增加了路径规划功能，这个功能将大大方便使用者，针对需要连续执行固定路径的场合，将需要连续执行的位置以表格的形式在向导中组态，只调用一个子例程即可完成复杂路径执行。

在实际情况中，很多时候工作的难点变成了如何获取想要绘制图案的坐标点位，对 S7-200 SMART PLC 的软件部分操作其他文档已有详尽的说明，本文档将重点介绍绘制图形的工作流，以绘制“SIEMENS” LOGO 的图案为例，逐步介绍如何使用各软件工具，最终获取可供 CPU 执行的坐标点位来完成绘制项目。

文档中涉及到除 STEP7 MicroWIN SMART 外多个软件的使用，本文档将只介绍与整个流程相关的操作，其他的部分如软件如何获取安装、软件对系统的兼容性要求、其他功能的专业操作等均不在该文档讨论之列。同时本文中介绍的方法只是绘制图案工作流程的其中一种，用户只需参照对应的流程即可。

要查看全部参考信息，请参阅 S7-200 SMART 系统手册，文档编号 109745610。下载链接如下：

<https://support.industry.siemens.com/cs/cn/zh/view/109745610>

2 要求及其他工具

2.1 PLC 软硬件要求

软件要求:

- STEP 7-MicroWIN SMART 软件版本:V2.7 及以上

硬件要求:

- 标准型 S7-200 SMART CPU 固件版本:V2.7 及以上

2.2 其他工具

Adobe Illustrator 2020 (矢量绘图软件)

AutoCAD 2023 (计算机辅助设计软件)

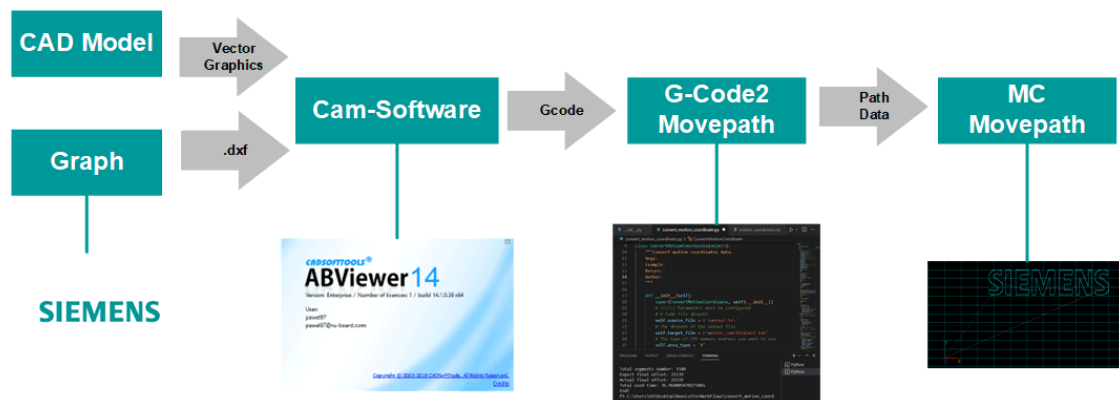
ABViewer14 (dxf to G-Code)

Visual Studio Code (跨平台源代码编辑器)

Python-3.9.13-amd64 (python 运行环境安装包)

注: 这里的工具软件仅作为参考不指定, 其他相关的软件也可以实现功能, 客户可根据自己擅长的工具酌情使用

3 workflow 概览



生成路径坐标信息的整个工作流程如图所示，熟悉 **Siemens Kinematics** 运动控制库的工程师对这个流程一定不陌生。**S7-200 SMART** 从运动控制支持方面也沿用了更高阶控制器的做法，操作流的一致性也使得控制器间的使用经验得以复用。该流程的本质涉及到三个重要的转换：

- 从位图图片（.PNG）到矢量图图片（.DXF）转换
- 矢量图到 **G-Code** 的转换
- **G-Code** 到 DB 块（XYZ 坐标）的转换

3.1 Vector Graphics

第一步转换，核心的难点在于图形矢量化。所谓矢量图，就是使用直线和曲线来描述的图形，构成这些图形的元素是一些点、线、矩形、多边形、圆和弧线等，它们都是通过数学公式计算获得的，具有编辑后不失真的特点。

如果以图片格式来说明的话，第一步的转换就是将如.bmp、.jpg、.png 等常见的格式图片转为.dwg、.dxf、.ai 等格式的矢量图形。这一步也是在所有流程中最重要的一步。当然，我们也可以直接使用如 **AutoCAD** 等矢量绘图工具直接绘制相关的图形文件，那么转换的步骤自然可以省略。

3.2 G-Code

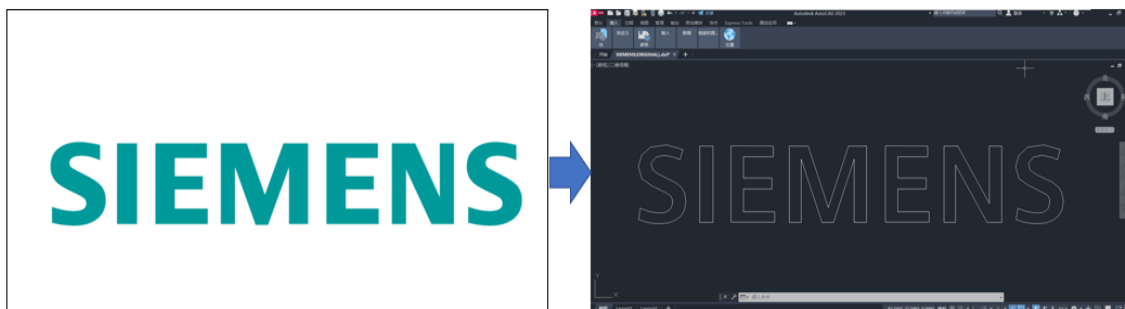
第二步转换，也就是矢量图到 **G-Code** 的转换，相对来说是从技术层面最容易实现的。这种应用场景在数控机床的加工中有大量成熟的应用案例，在线搜索“**dxf to GCode**”就可以找到一些免费的生成工具来实现这部分的转换。本文档中使用 **ABViewer** 这个工具进行操作说明。

3.3 PathData

第三步转换，也就是从 **G-Code** 到 **PLC** 可执行的坐标数据之间的转换，因为每个厂家的不同定位的 **CPU** 可以执行的数据块都不尽相同，**Kinematics** 运动控制库中提供的小工具生成的 **DB** 块也不能在用 **SMART PLC** 的软件平台，所以目前并没有通用的工具用来执行该转换。针对这种情况，西门子开发了基于 **Python** 语言的例子程序，将 **G-Code** 转换成 **SMART PLC** 可用的数据块以供参考使用。

4 获取矢量图文件

在实际操作的过程中，如果以获得图案坐标点位为最终目的，获取CAD格式文件无疑是最重要的一步，因为这涉及从位图图片到矢量图图片转换的技术难点。本文中介绍的转换操作方法基于Adobe illustrator软件。实际上，至今也没有任何一款软件可以完美的把复杂的位图照片转成矢量图，假如位图与矢量图可以完美转换，由于矢量图缩放不失真文件体积小且携带大量信息的特性，我们平常使用的图片将很快都切换成矢量图形式。所以本文档介绍的流程部分目前只针对简单图形转换的应用场景，同时仍需要转换之后的编辑与加工才能保证效果，想要完全靠软件一键转换是不现实的。



4.1 Adobe Illustrator 简述

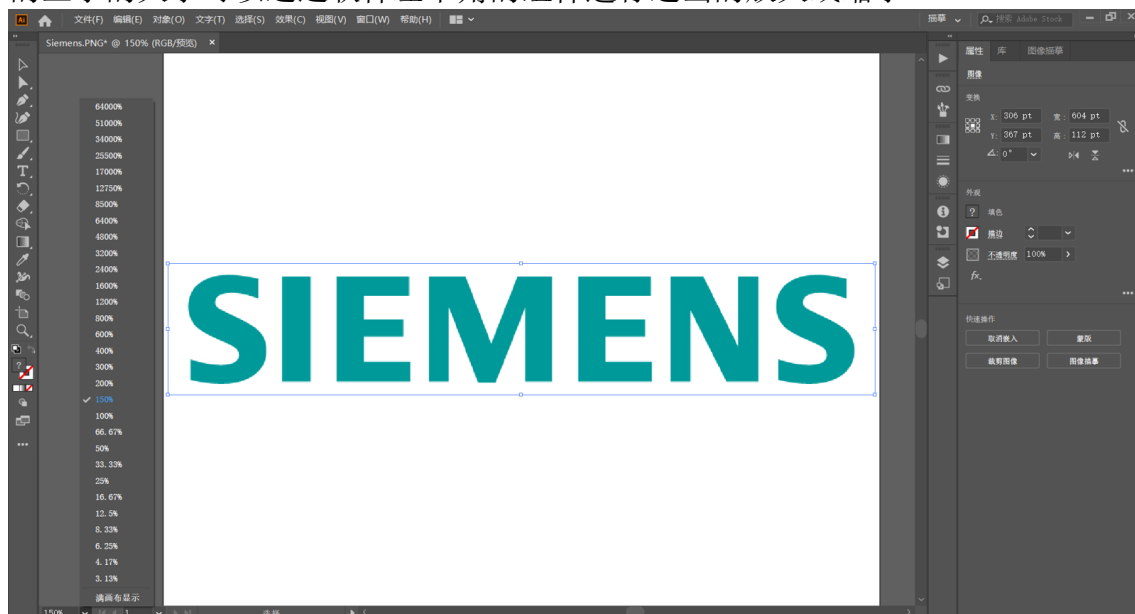


Adobe illustrator，简称 AI，是一种应用于出版、多媒体和在线图像的工业标准矢量插画软件。它是一款专业图形设计工具，提供丰富的像素描绘功能以及顺畅灵活的矢量图编辑功能，在本文档中将使用这款软件完成照片矢量化流程。

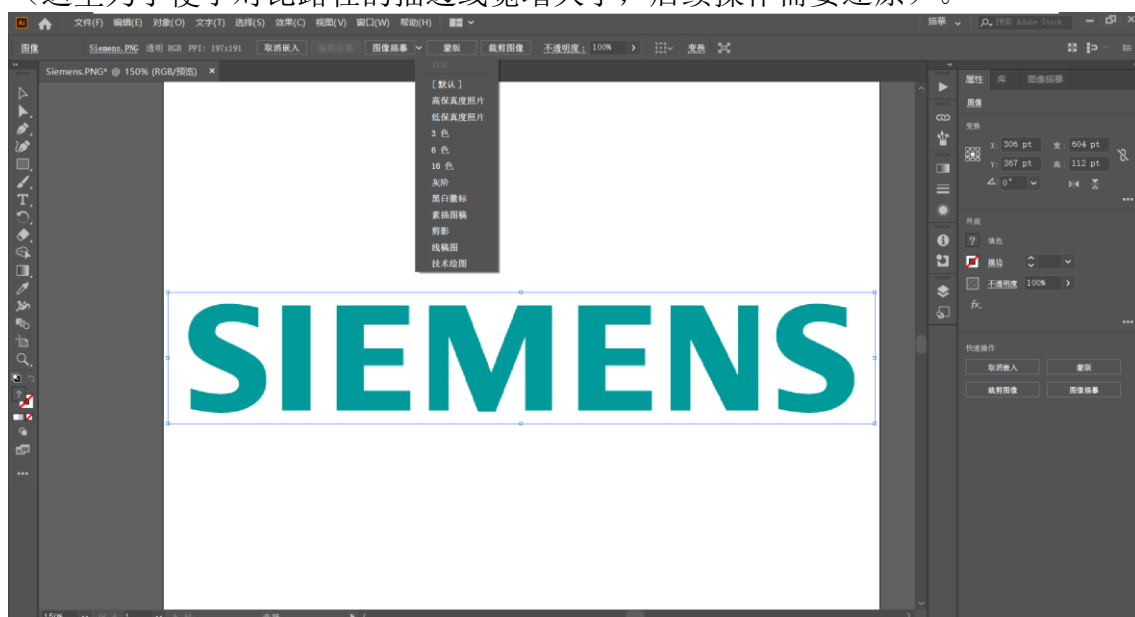
需要说明的是，由于 Adobe illustrator 高阶版本对电脑的配置比较高，同时各版本的软件同样具备相同的图片矢量化的功能，所以文档本次以 Adobe illustrator 2020 版本为例，为大家介绍整个软件与矢量化相关的操作。

4.2 操作步骤

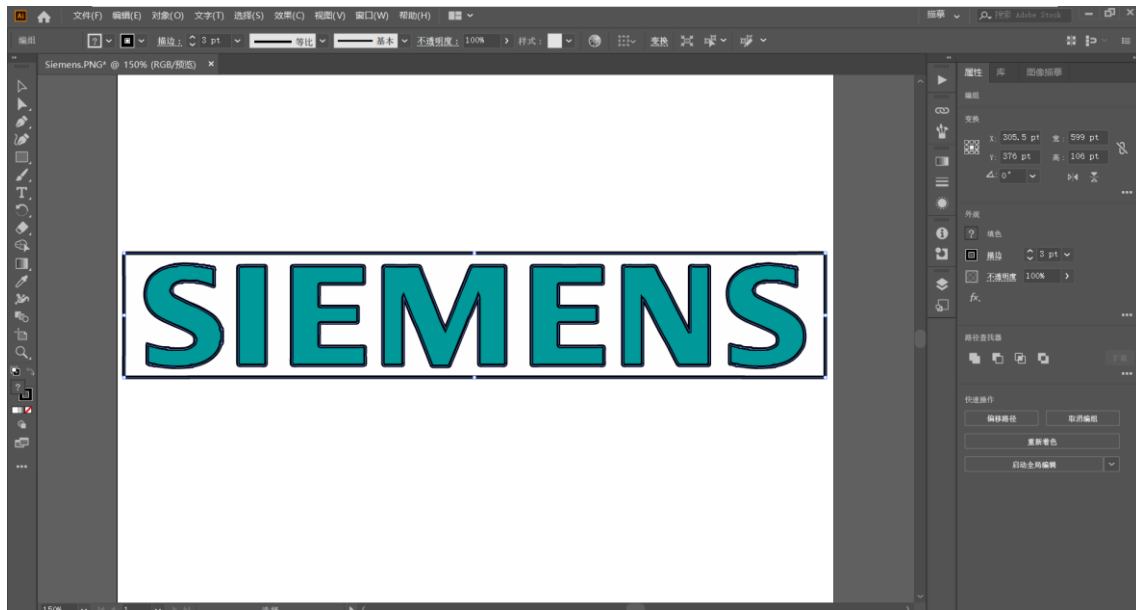
首先打开 Adobe Illustrator 软件，在软件的顶部的工具栏点击“文件”-“打开”，在弹出的对话框中选择存放图片的文件夹并选择对应的图片打开。在打开图片后图片的比例和位置可能不合适，手动将图片调整至编辑版面中心即可，同时画布的显示的大小可以通过软件左下角的组件进行适当的放大或缩小。



在软件的顶部的工具栏点击“窗口”-勾选“控制”选项卡，此时在工具栏下方将会再出现一排快捷栏的栏位。选中图片后在快捷栏位的“图像描摹”下拉箭头的选项栏中，选择“低保真度照片”。选择完成后系统会处理图像，然后在快捷箭头的右侧会出现一个“扩展”的按钮，点击该按钮即可为图片添加边缘路径（这里为了便于对比路径的描边线宽增大了，后续操作需要还原）。



4 获取矢量图文件

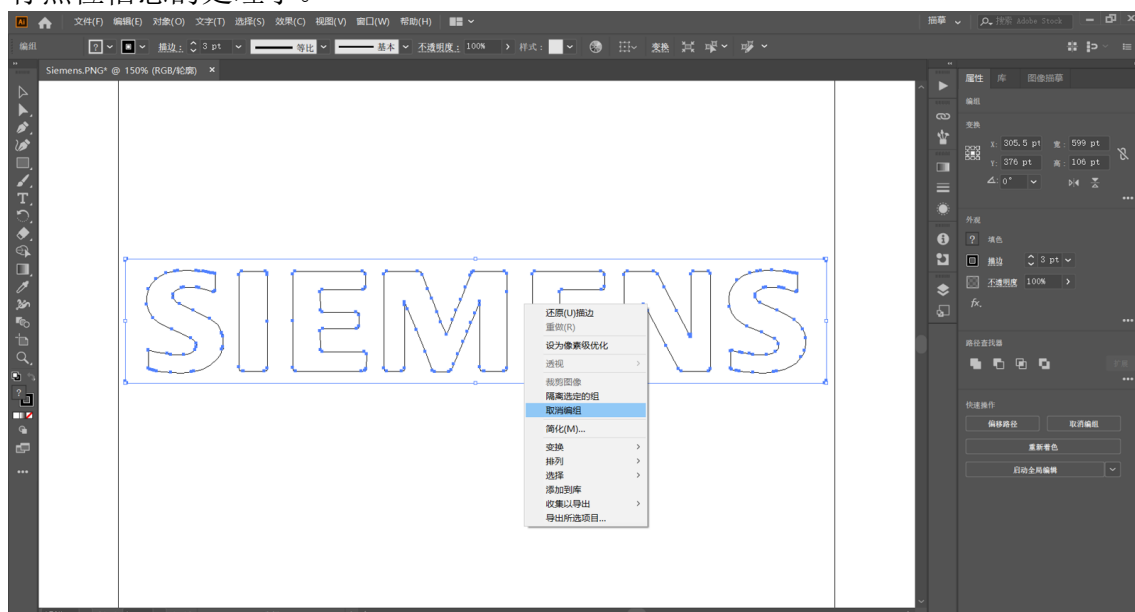


在软件的顶部的工具栏点击“视图”——选择“轮廓”模式，此时整个画布将只显示边缘轮廓。

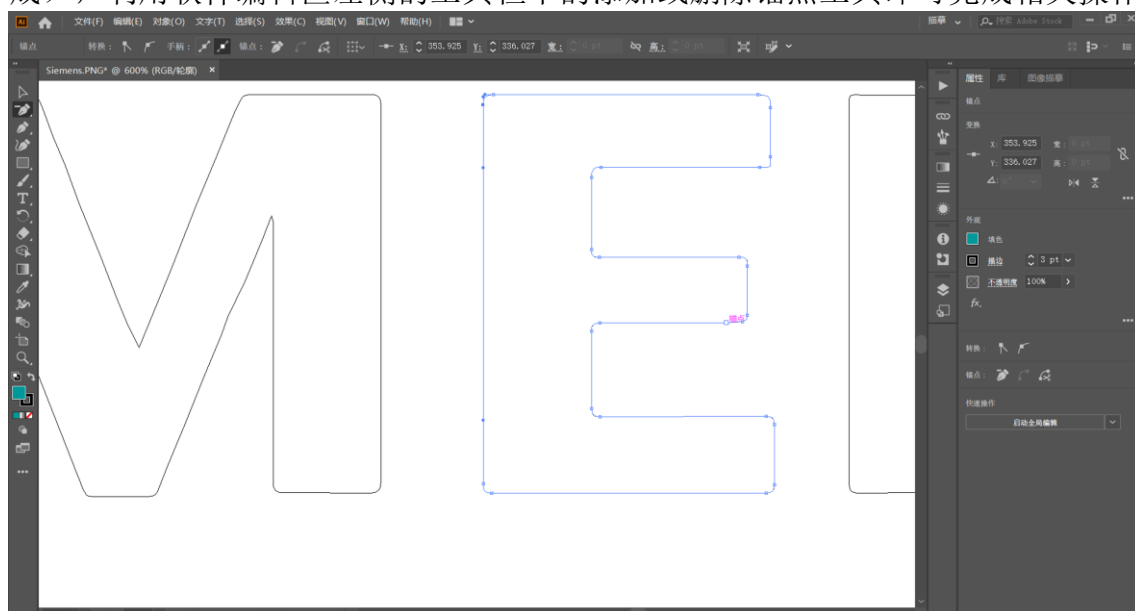


4 获取矢量图文件

由于画面中各自连续闭合的曲线都相当于是一个整体不便于修改，选择全部轮廓信息后，单击鼠标右键选择“取消编组”，设定完成后就可以对每个单独字符进行点位信息的处理了。

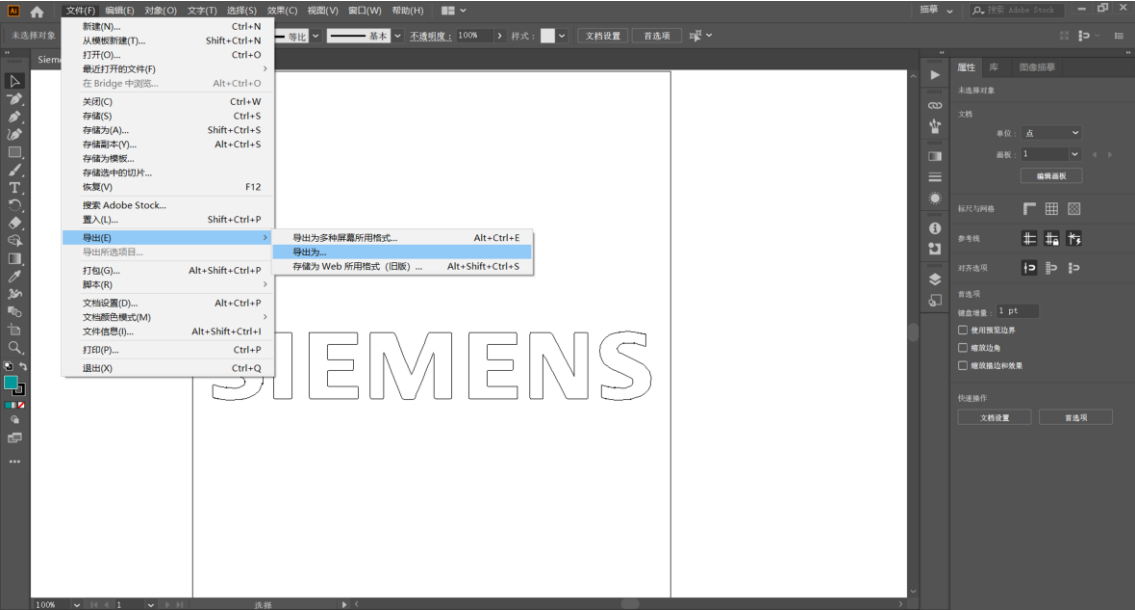


由于后续转换的 G-Code 的复杂度与原始文件的锚点数量有关，受制于 SMART PLC 路径规划功能最多 128 条线段的限制，前期处理尽量选择更少的点位信息来生成矢量图文件，所以在当前步骤中应尽量在保证图形信息完整的情况下，删除多余的锚点位，如一条直线中的多个锚点位（此步骤也可以在 AutoCAD 中完成），利用软件编辑区左侧的工具栏中的添加或删除锚点工具即可完成相关操作。

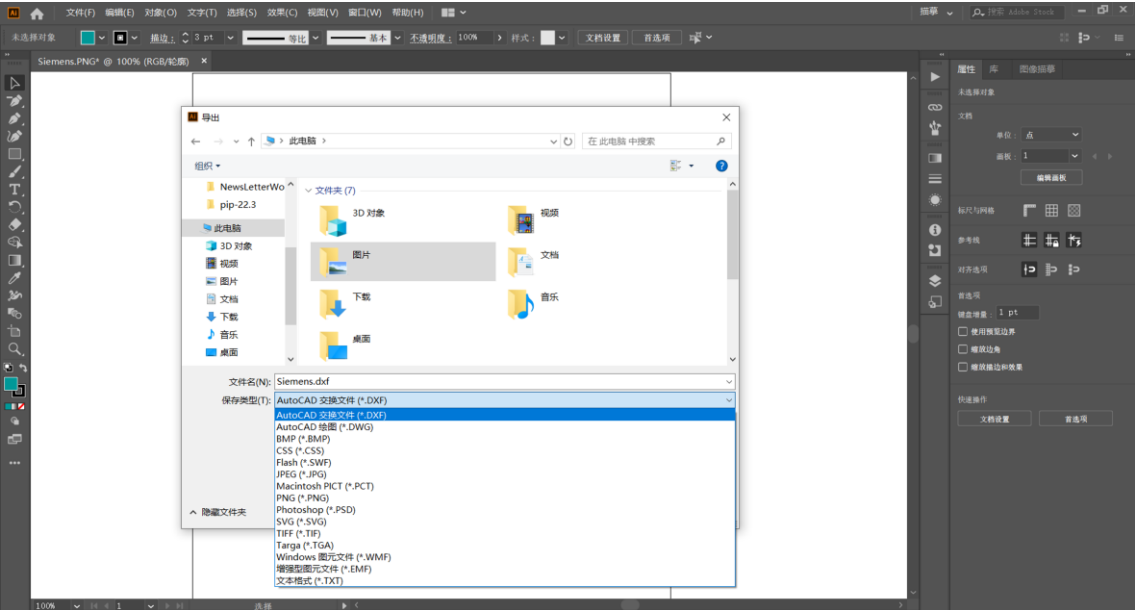


调整好锚点数量后即可将文件导出，在软件的顶部的工具栏点击“文件”--“导出”--“导出为”来选择将文件导出的格式。

4 获取矢量图文件

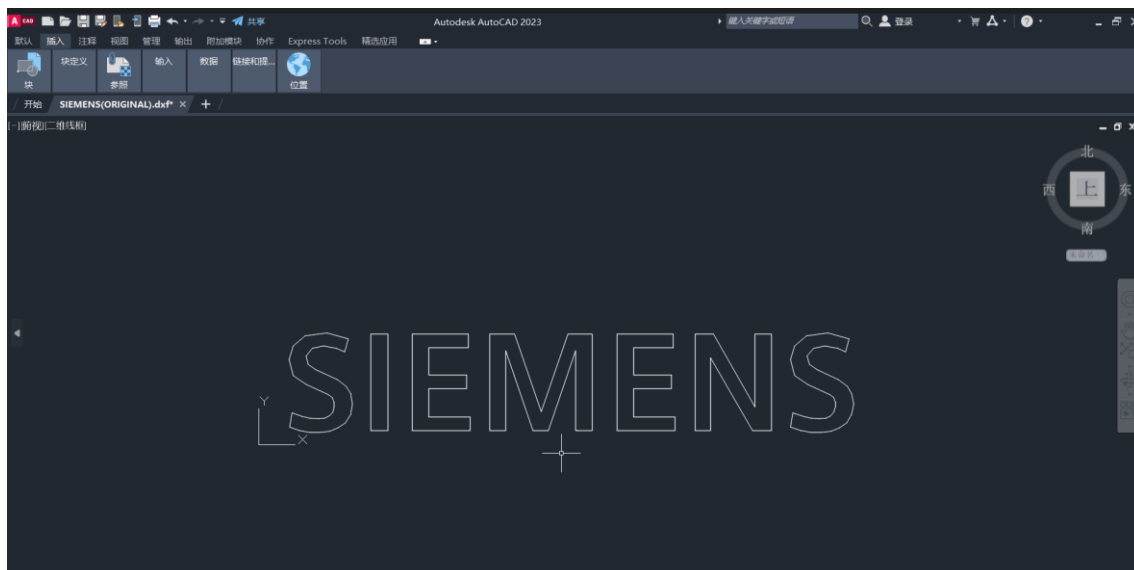


在弹出的对话框中可以选择导出文件存放的目标文件夹以及导出格式，Adobe illustrator 软件提供了多种格式供客户选择，这里推荐导出.DXF 格式的 AutoCAD 交换文件，方便后续的编辑使用。



将文件导出后，使用 AutoCAD 软件仍需要编辑，具体编辑部分不赘述，最终获得的矢量图文件效果如下图所示，至此，基于图片量化的第一部分转换全部完成。

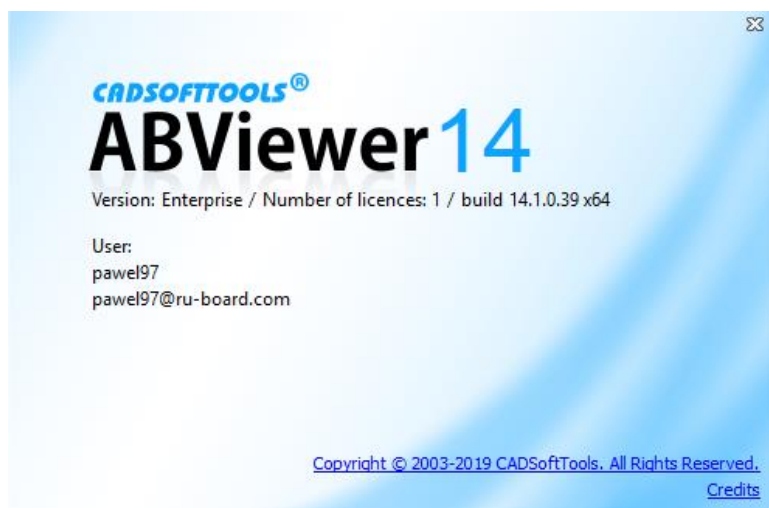
4 获取矢量图文件



5 获取 G-Code

当我们获得了.DXF 格式的矢量图文件，获得 G-Code 就方便了很多，在线搜索“dxf to GCode”就可以找到一些免费的生成工具。本文档中使用 ABViewer 这个工具进行操作说明。

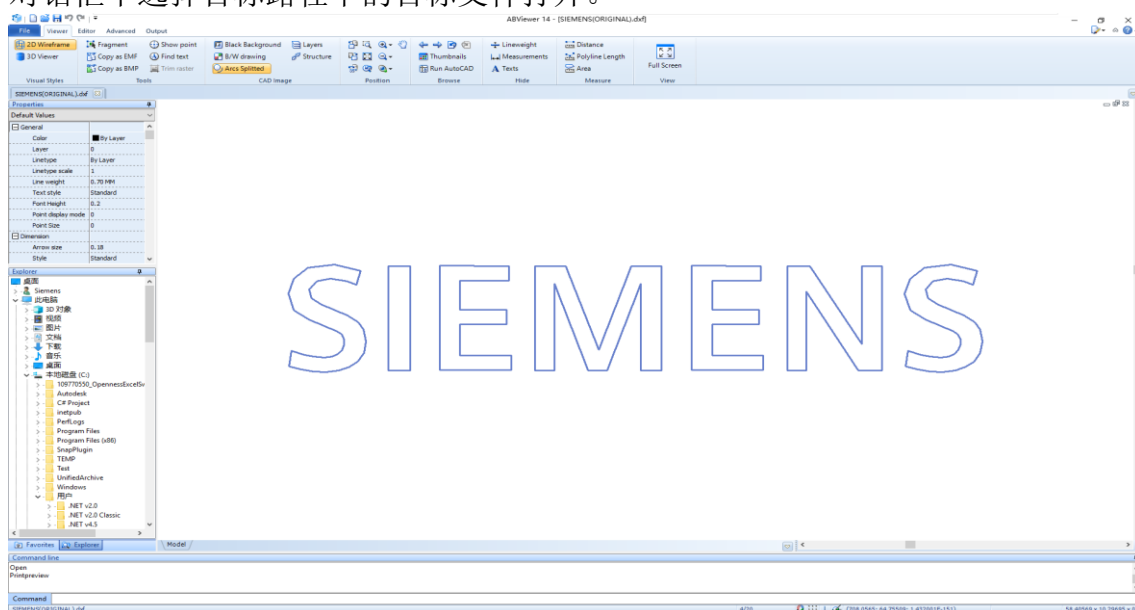
5.1 ABViewer 简述



ABViewer 是一款高品质、低成本、高效率的多功能设计及工程文档管理和图像浏览工具，程序具有批处理功能，是专业的 2D/3D CAD 查看器、编辑器和转换器。软件同时支持 30 多种光栅和矢量图形格式，并可以精确的调整图像或转换其它文件格式。本文档主要使用到了该软件 CAD to G-Code 的功能。

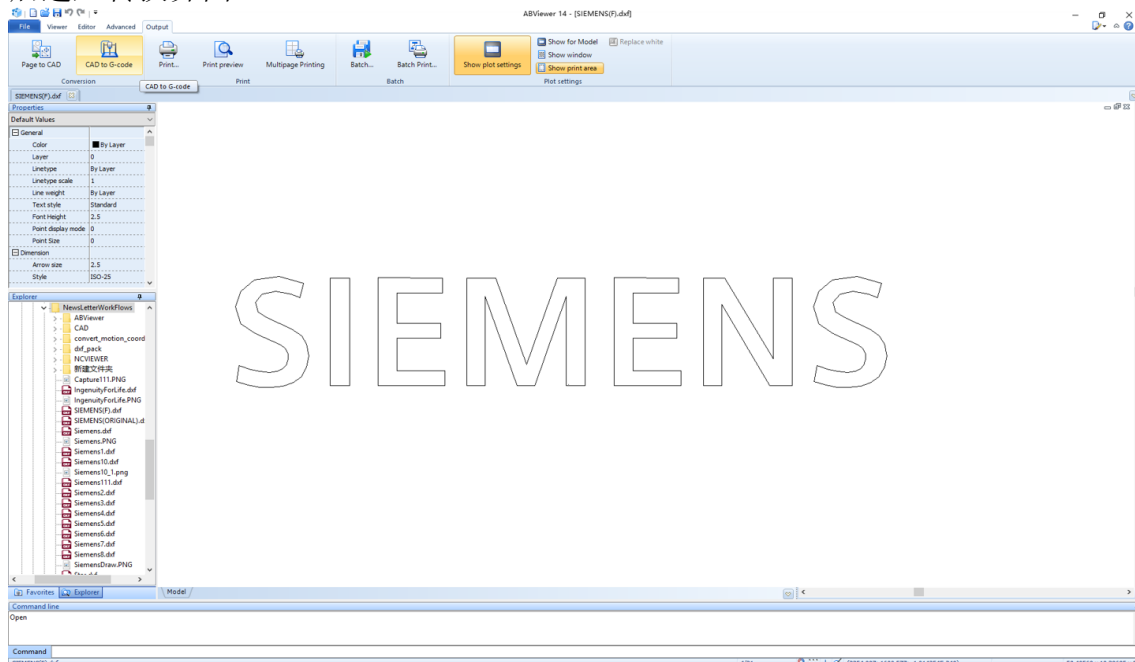
5.2 操作步骤

首先打开 ABViewer 软件，在软件上方的工具栏“File”--“Open”，在弹出的对话框中选择目标路径下的目标文件打开。

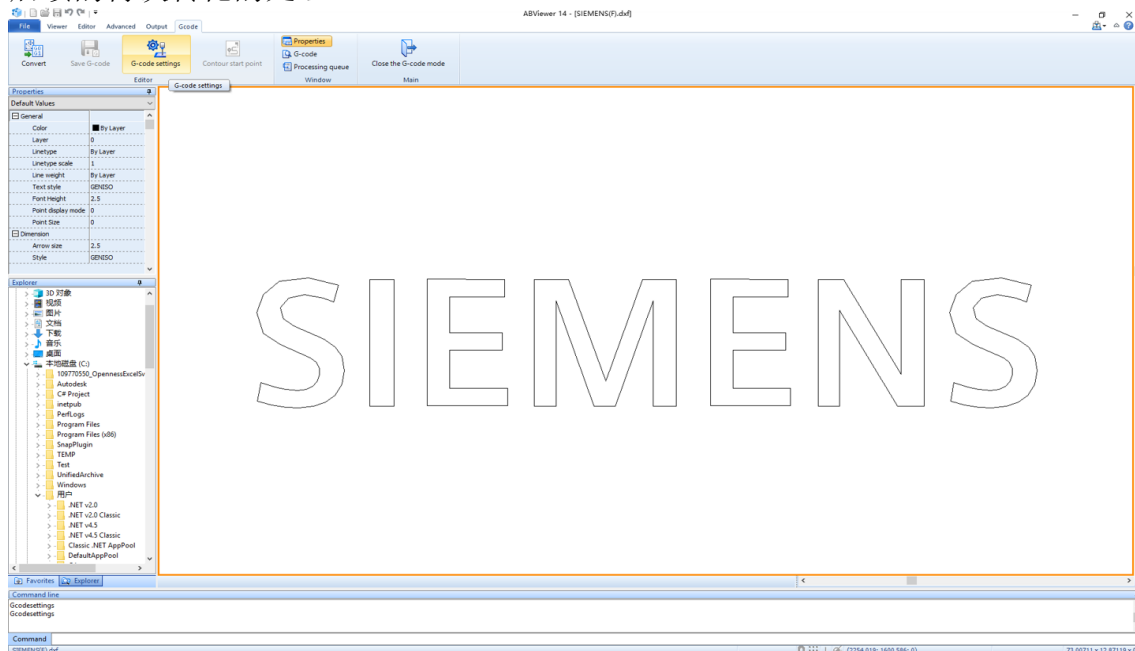


5 获取 G-Code

打开文件以后，在软件上方的工具栏“Output”--“CAD to G-Code”鼠标单击后进入转换界面。



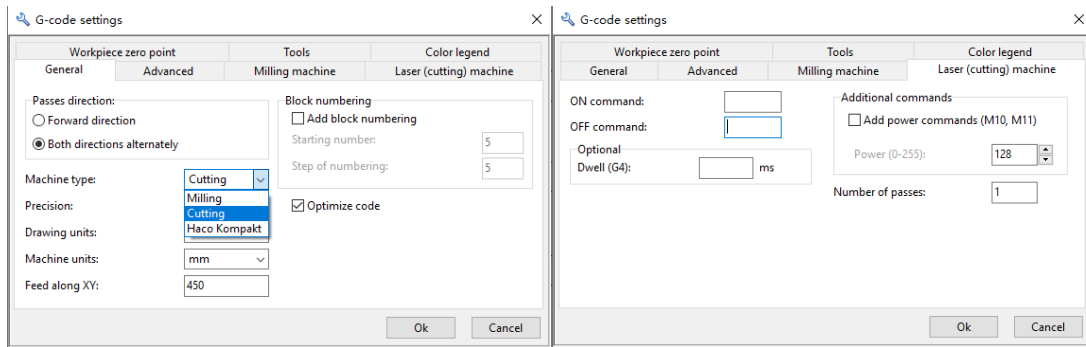
转换 G-Code 之前，在软件中可以设置一些与转换相关的参数，鼠标单击软件上方工具栏“G-Code setting”可在弹出的对话框中设定转换参数，因为后续的转换我们的 Python 代码只支持 G00 和 G01 等简单的 G-Code，如果不更改设定而是以默认参数生成的话，转化的 G-Code 可能包含复杂的数控刀具指令，不利于后续的再次转化的处理。



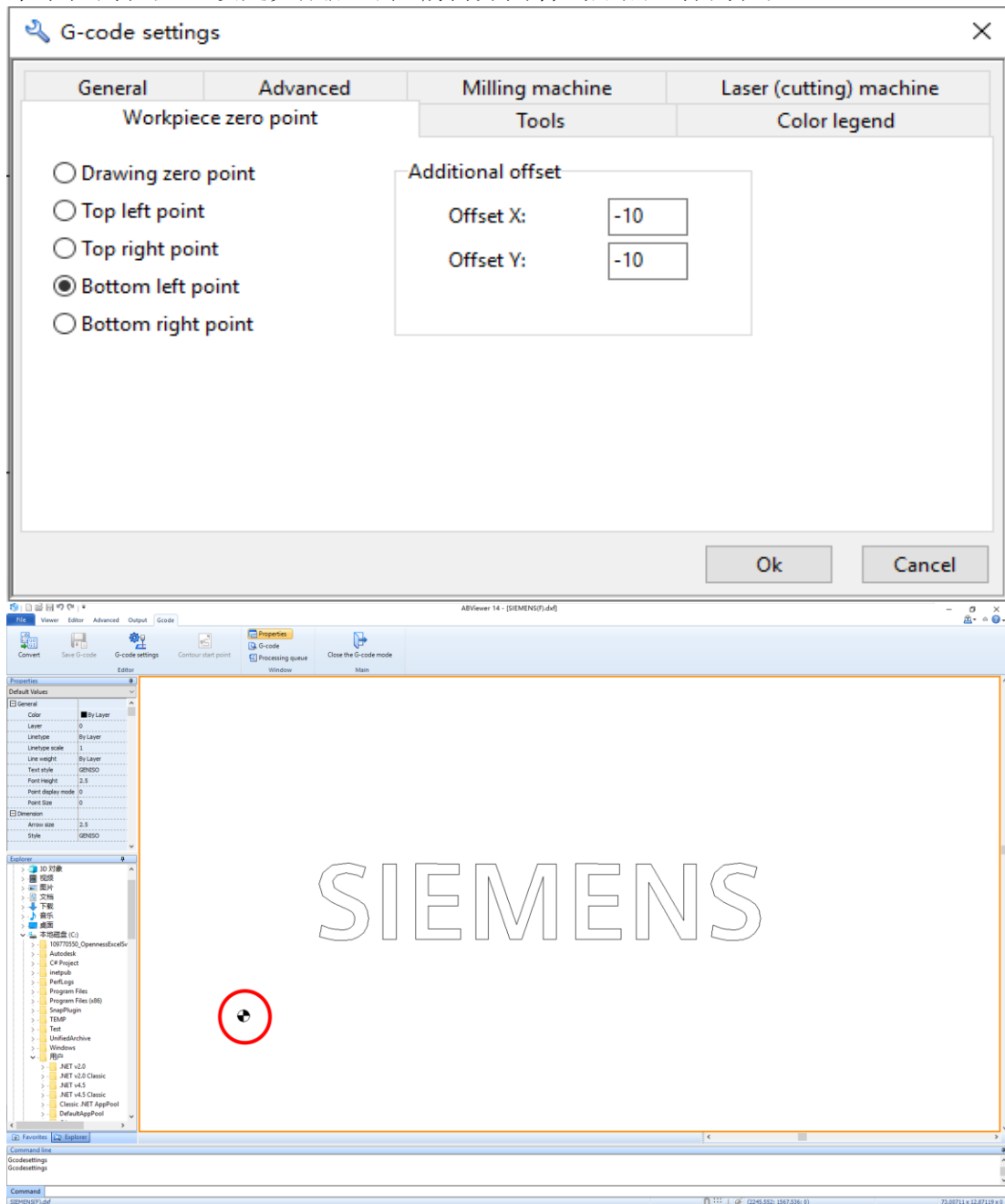
G-Code setting 部分需要设定的部分主要有两处：

- “General”选项卡下，“Machine type”选择“cutting”；
- “Laser (cutting) machine”选项卡下，ON/OFF command 文本框中的内容全部删除。

5 获取 G-Code



在“Workpiece zero point”选项卡下，可以设定 G-Code 的加工原点和原点偏移，在此例程中，加工原点选择整体图案的左下角在分别沿 XY 轴负方向偏移 10 个单位的位置，设定完成后可在编辑界面看到原点坐标的位置。



[illegible]

SIMATIC S7-200 SMART 直线插补样例一—绘制图像 workflow
V1.0 01/2023

6 获取路径坐标点

当获得 G-Code 文档后，便可以开始第三步的转换，也就是从 G-Code 到 PLC 可执行的坐标数据之间的转换。针对目前 SMART PLC 支持的数据块的情况，西门子开发了基于 Python 编程语言的程序，将 G-Code 转换成 SMART PLC 可用的数据块以供参考使用。但是在 Windows 系统中想要运行 Python 语言编写的程序需要满足安装环境才可以，本章节先简单说明 Python 运行环境搭建的相关步骤，再介绍运行程序来获取坐标点的操作。

6.1 Python 安装环境

Python 并不是一门新的编程语言，1991 年就发行了第一个版本，2010 年以后随着大数据和人工智能的兴起，Python 又重新焕发出了耀眼的光芒。在 2019 年 12 月份世界编程语言排行榜中，Python 排名第三，仅次于 Java 和 C 语言。

Python 是一门开源免费的脚本编程语言，它不仅简单易用，而且功能强大。它的运行环境安装包可以免费在官网获得，Python 安装包下载地址：

<https://www.python.org/downloads/>

打开该链接，可以看到有各个版本的 Python，如下图所示

Looking for a specific release?
Python releases by version number:

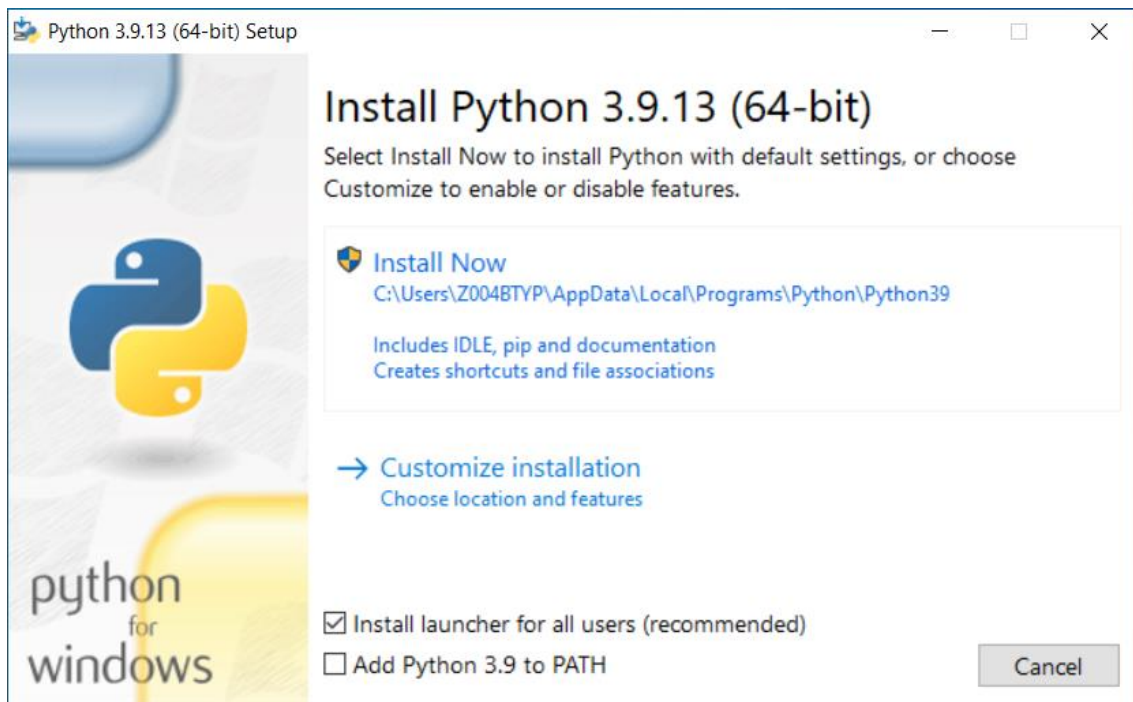
Release version	Release date		Click for more
Python 3.11.1	Dec. 6, 2022	Download	Release Notes
Python 3.10.9	Dec. 6, 2022	Download	Release Notes
Python 3.9.16	Dec. 6, 2022	Download	Release Notes
Python 3.8.16	Dec. 6, 2022	Download	Release Notes
Python 3.7.16	Dec. 6, 2022	Download	Release Notes
Python 3.11.0	Oct. 24, 2022	Download	Release Notes
Python 3.9.15	Oct. 11, 2022	Download	Release Notes

这里不推荐使用最新版本，较新的版本可能对电脑的配置兼容性要求都比较高，以安装 Python3.9.13 的版本为例，点击上图中的版本号或者“Download”按钮进入对应版本的下载页面，滚动到最后即可看到各个平台的 Python 安装包，选择对应系统的安装包即可。

Files

Version	Operating System	Description	MD5 Sum	File Size	GP
Gzipped source tarball	Source release		eafda83543bad127cadef4d288fdab87	26355887	SIG
XZ compressed source tarball	Source release		5e2411217b0060828d5f923eb422a3b8	19754368	SIG
macOS 64-bit Intel-only installer	macOS	for macOS 10.9 and later, deprecated	671848930809decf27f586ddf98c6e9b	30997161	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	76b63cf623e32cdf27c5033434bd69ce	38821163	SIG
Windows embeddable package (32-bit)	Windows		fec0bc06857502a56dd1aeeae6488ef8	7729405	SIG
Windows embeddable package (64-bit)	Windows		57731cf80b1c429a0be7133266d7d7cf	8570740	SIG
Windows help file	Windows		c86feba059b340a1de2a9d2ee7059a6d	8953644	SIG
Windows installer (32-bit)	Windows		46c35b0a2a4325c275b2ed3187b08ac4	28096840	SIG
Windows installer (64-bit)	Windows	Recommended	e7062b85c3624af82079794729618eca	29235432	SIG

下载完成后点击安装包，进入安装进程



请尽量勾选“add Python3.9 to PATH”这样可以将 Python 命令工具所在目录添加到系统 Path 环境变量中，以后开发程序或者运行 Python 命令会非常方便。安装过程不赘述，用户可根据自己需要的功能和安装路径设定完成安装。更多详细信息可参照该教程：<http://c.biancheng.net/view/4161.html>

安装完成以后，打开 Windows 的命令程序（命令提示符），在窗口中输入命令 `python --version`，如果出现 Python 的版本信息，就说明安装成功了，如下图所示。

```
C:\Windows\system32\cmd.exe
```

```
Microsoft Windows [版本 10.0.17763.3770]
(c) 2018 Microsoft Corporation。保留所有权利。
```

```
C:\Users\G4>python --version
Python 3.9.13
```

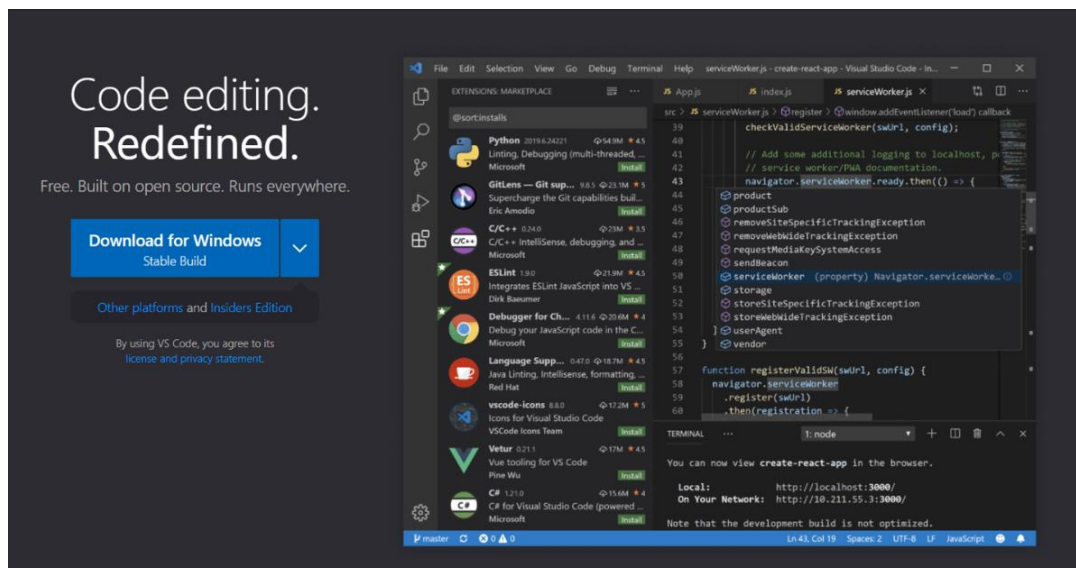
```
C:\Users\G4>
```

安装成功后，便可以进行下一步的运行 Python 代码生成坐标点的文档了。

6.2 Python 代码运行

与本文档搭配使用的 Python 例子程序需要在源代码编辑器中运行，文档中以 Visual Studio Code 为例来演示操作步骤。

6.2.1 Visual Studio Code 简述

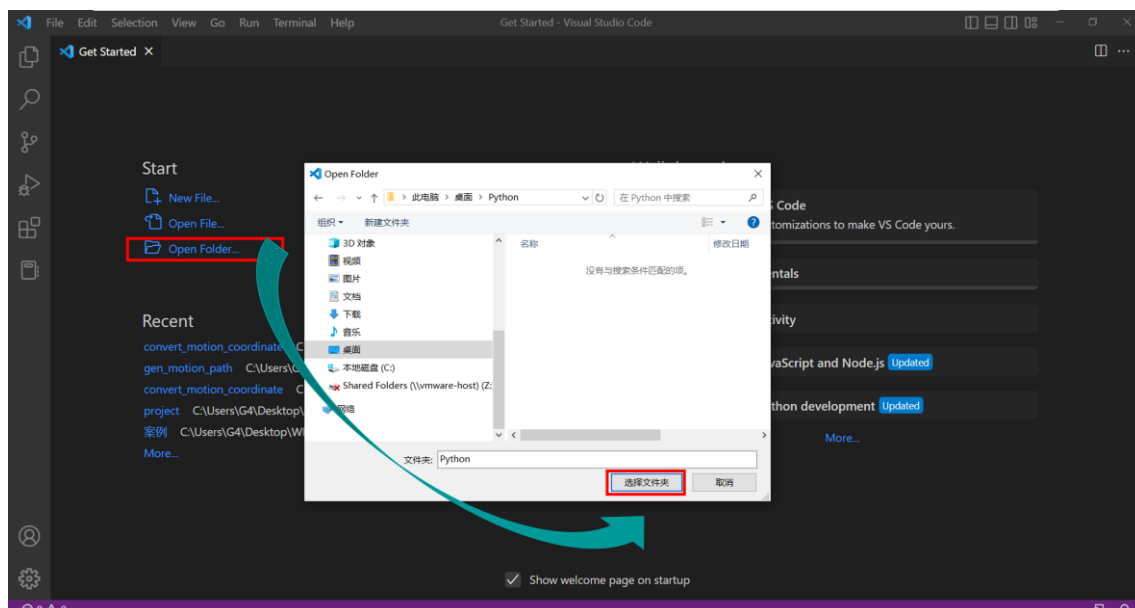


Visual Studio Code（简称“VS Code”）是 Microsoft 在 2015 年 4 月 30 日 Build 开发者大会上正式宣布一个运行于 Mac OS X、Windows 和 Linux 之上的，针对于编写现代 Web 和云应用的跨平台源代码编辑器，可在桌面上运行，并且可用于 Windows，macOS 和 Linux。它具有对 JavaScript，TypeScript 和 Node.js 的内置支持，并具有丰富的其他语言（例如 C++，C#，Java，Python，PHP，Go）和运行时（例如 .NET 和 Unity）扩展的生态系统，是各种源代码开发人员手中的利器。

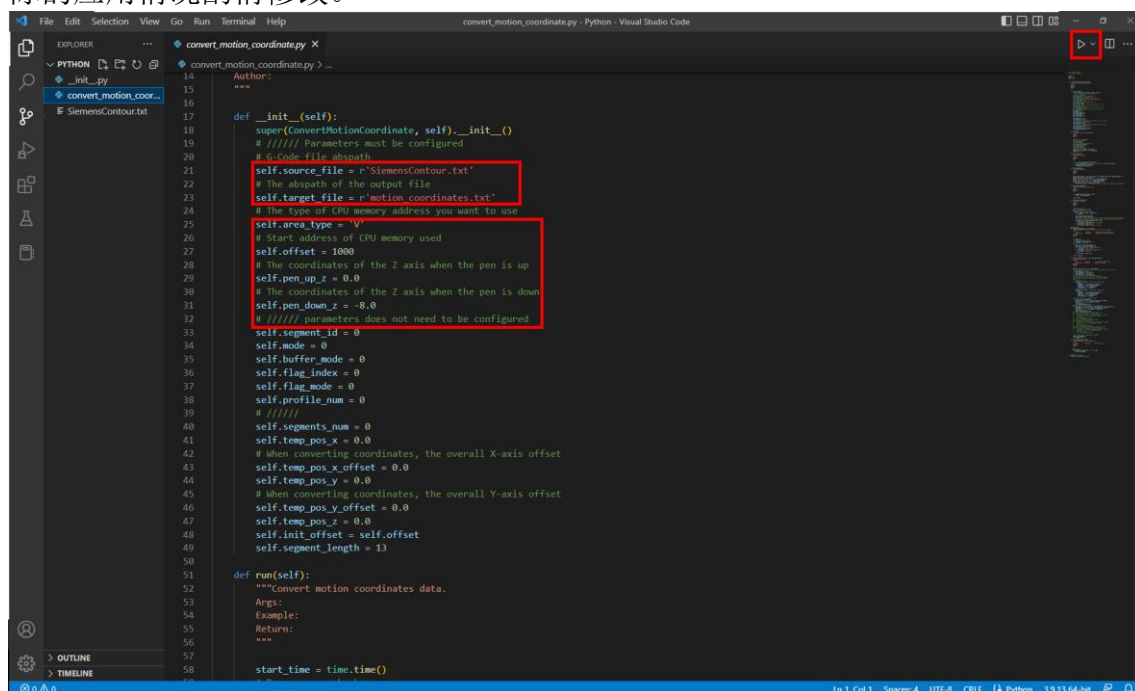
6.2.2 程序执行

运行程序之前，将与文档配套的程序和在第五章生成的.txt 格式的 G-Code 文件同时放入一个文件夹中，打开 VS Code 软件，选择“Open Folder”，在弹出的窗口中找到对应路径，选择已放置好所有文档的文件夹。

6 获取路径坐标点

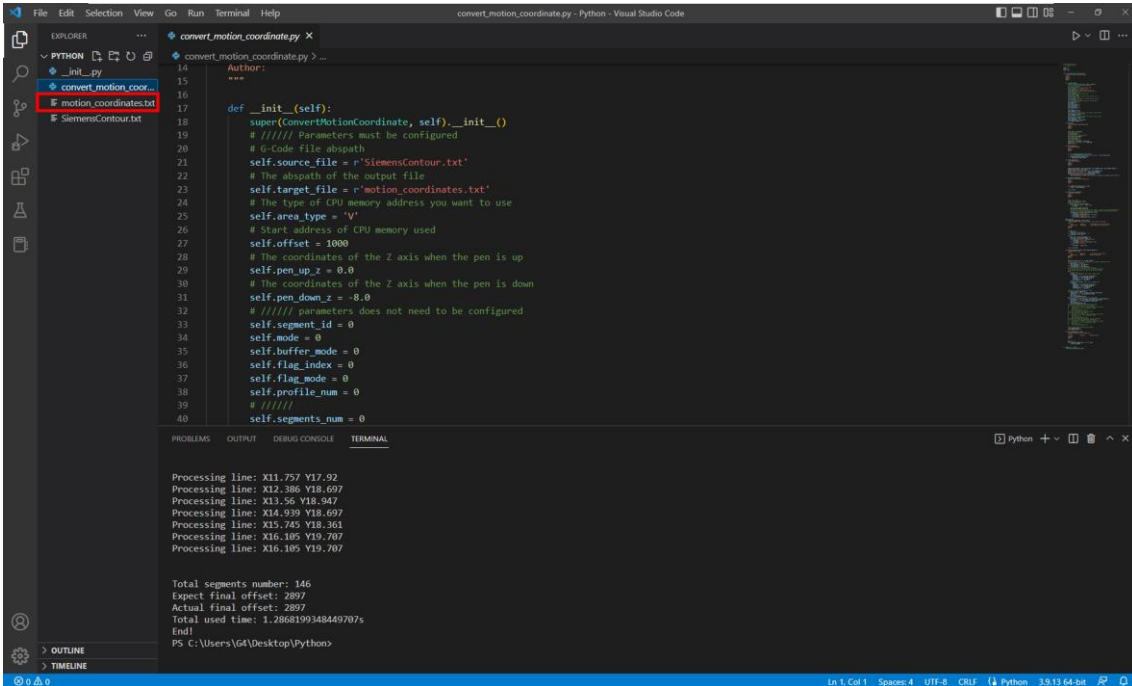


选择信任所选中的文件夹后，即可进入到程序的编辑显示界面，该例子程序主要的功能就是将 G-Code 文件转换成.txt 格式的 SMART PLC V 寄存器区坐标数据文件。程序的具体逻辑不赘述，需要说明的是下图中红色方框圈住的区域，第一个方框声明的是源 G-Code 文件和输出坐标文件的名称，第二个方框声明的是 V 区写入数据的起始地址和 Z 轴运动时抬笔和落笔的坐标数据。客户可根据自己实际的应用情况酌情修改。



确认程序无误不需其他修改后，鼠标单击软件的右上角小三角箭头按钮，即可运行程序。

6 获取路径坐标点



```
def __init__(self):
    super(ConvertMotionCoordinate, self).__init__()
    # Parameters must be configured
    # G-Code file abs path
    self.source_file = r'SiemensContour.txt'
    # The abs path of the output file
    self.target_file = r'motion_coordinates.txt'
    # The type of CPU memory address you want to use
    self.area_type = 'V'
    # Start address of CPU memory used
    self.offset = 1000
    # The coordinates of the Z axis when the pen is up
    self.pen_up_z = 0.0
    # The coordinates of the Z axis when the pen is down
    self.pen_down_z = -8.0
    # Parameters does not need to be configured
    self.segment_id = 0
    self.mode = 0
    self.buffer_mode = 0
    self.flag_index = 0
    self.flag_mode = 0
    self.profile_num = 0
    # Parameters does not need to be configured
    self.segments_num = 0
```

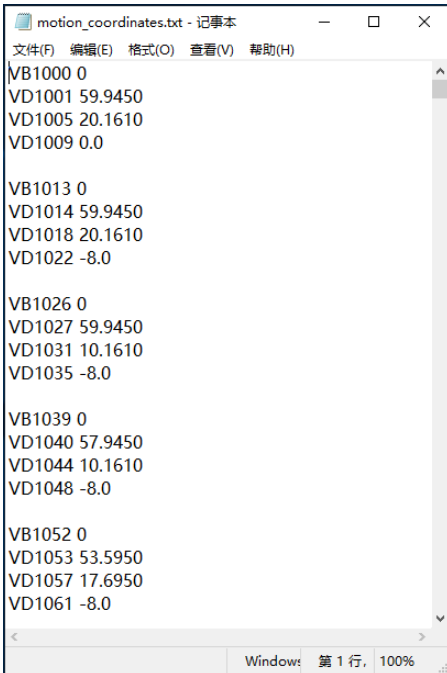
```
Processing line: X11.757 Y17.92
Processing line: X12.386 Y18.697
Processing line: X13.56 Y18.947
Processing line: X14.939 Y18.697
Processing line: X15.745 Y18.361
Processing line: X16.105 Y19.707
Processing line: X16.105 Y19.707

Total segments number: 146
Expect final offset: 2897
Actual final offset: 2897
Total used time: 1.2868139348449707s
End!
```

程序运行结束后，软件下方的编译执行信息无报错，可以看到程序项目树区域新增加了一个名为“motion_coordinates.txt”的文件，也就是程序中提前声明好的输出文件，该文件的生成路径，即是整个项目文件夹的路径。至此，基于 Python 程序的第三部分转换全部完成。

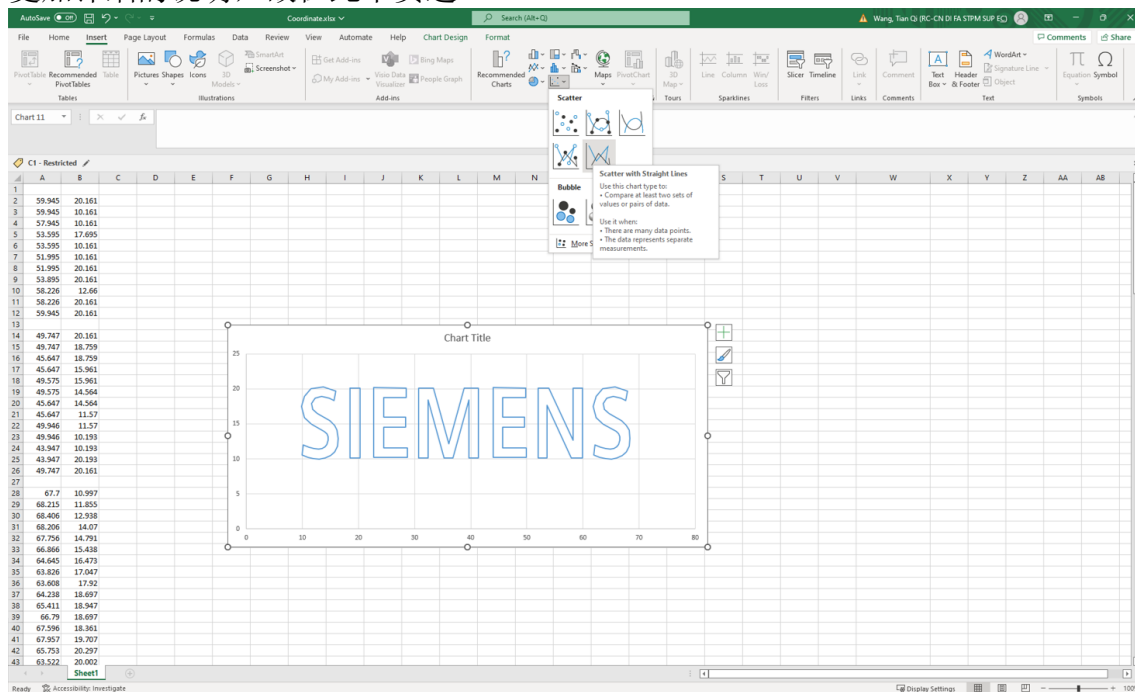
6.2.3 执行结果

在项目文件夹下打开生成的文件，可看到最终可供 PLC 程序执行的坐标点位信息（如下图所示一部分点位）。因为在 Python 程序中 V 区的起始地址值设置为 1000，所以第一个点位的信息也是从 VB1000 开始记录。



6 获取路径坐标点

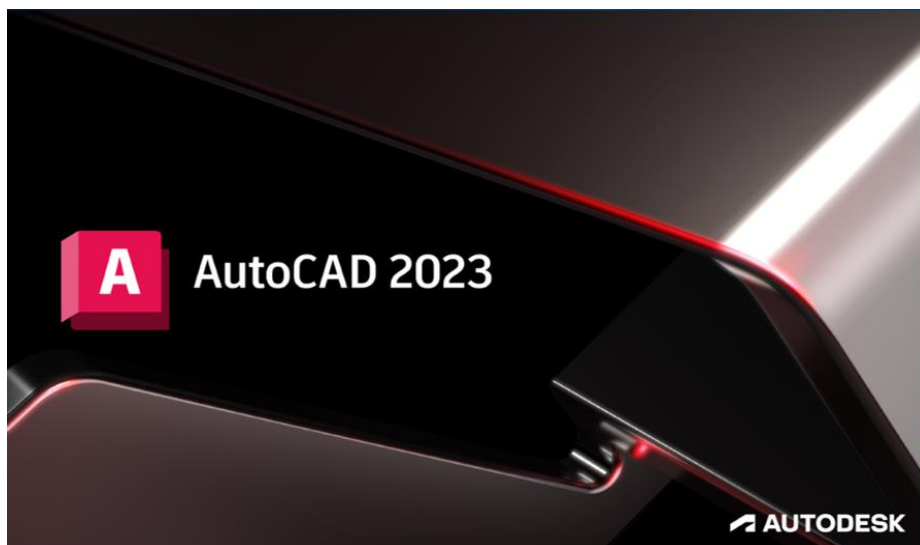
将所有的点位信息复制到 **Excel** 表格中，使用插入散点图的方式即可验证点位信息的正确与否。同时在 **Excel** 表格中可以直接将数据复制到 **SMART** 编程软件运动控制向导的表格中，通过 **MovePath** 指令来执行程序，这部分操作其他文档有更加详细的说明，故在此不赘述。



7 简单的坐标获取方法

前面的章节中我们已经介绍了通过三步转换法这种相对复杂的获取坐标点的过程，实际上当我们获得了想要加工项目的 CAD 文件后，如果图像的线段数量较少且方便记录的话，我们可以利用 AutoCAD 软件本身的软件特性轻易的获得各个目标点位。

7.1 AutoCAD 简述



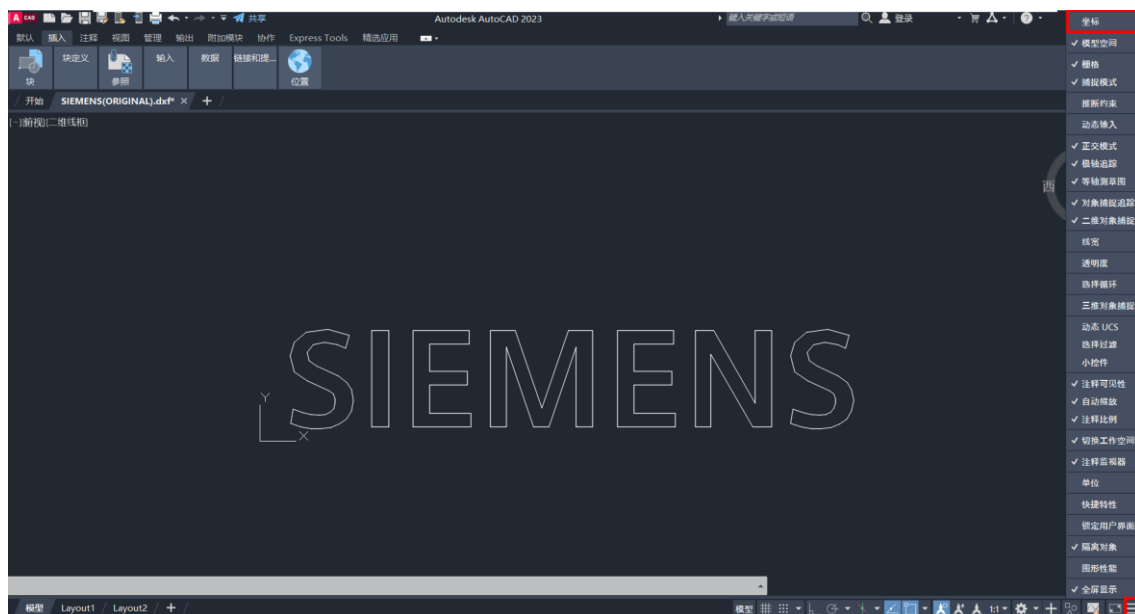
AutoCAD（Autodesk Computer Aided Design）是 Autodesk（欧特克）公司首次于 1982 年开发的自动计算机辅助设计软件，用于二维绘图、详细绘制、设计文档和基本三维设计，现已经成为国际上广为流行的绘图工具。AutoCAD 具有良好的用户界面，通过交互菜单或命令行方式便可以进行各种操作。它的多文档设计环境，让非计算机专业人员也能很快地学会使用。在不断实践的过程中更好地掌握它的各种应用和开发技巧，从而不断提高工作效率。AutoCAD 具有广泛的适应性，它可以在各种操作系统支持的微型计算机和工作站上运行。

AutoCAD 有多个版本，本文以 AutoCAD 2023 版本为例，为大家介绍简易的坐标获取方法。

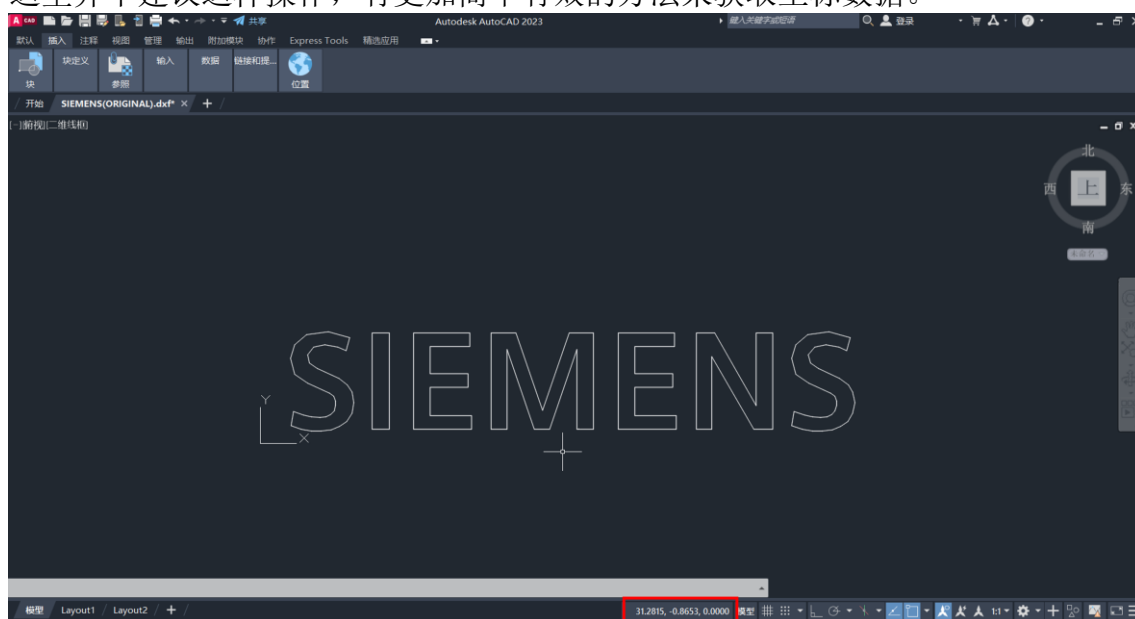
7.2 操作步骤

仍然以绘制“SIEMENS” LOGO 图案为例，当我们获得矢量图格式的文件时，使用 AutoCAD 打开文档进入编辑区后，点击软件右下方的配置按钮，勾选坐标选项卡。

7 简单的坐标获取方法

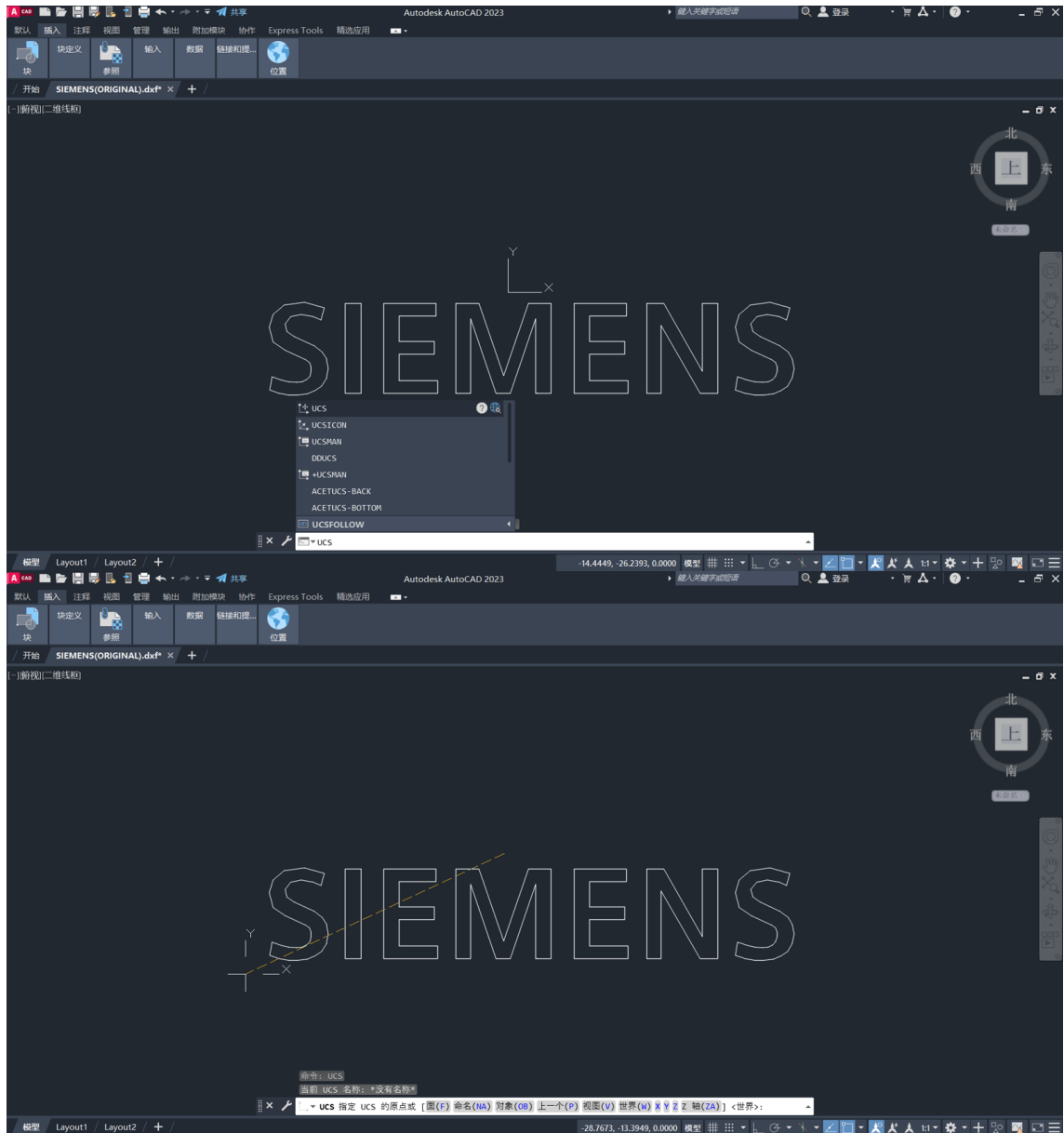


这样操作之后，当我们的鼠标在 CAD 软件的编辑区任意的移动时，软件底部的信息栏位中就会针对当前的光标点给出实时的坐标反馈，如果精度要求不高的场合，可以将鼠标指针分别放大移动到对应线段点位的端点处来分别记录坐标点位，但这里并不建议这样操作，有更加简单有效的方法来获取坐标数据。



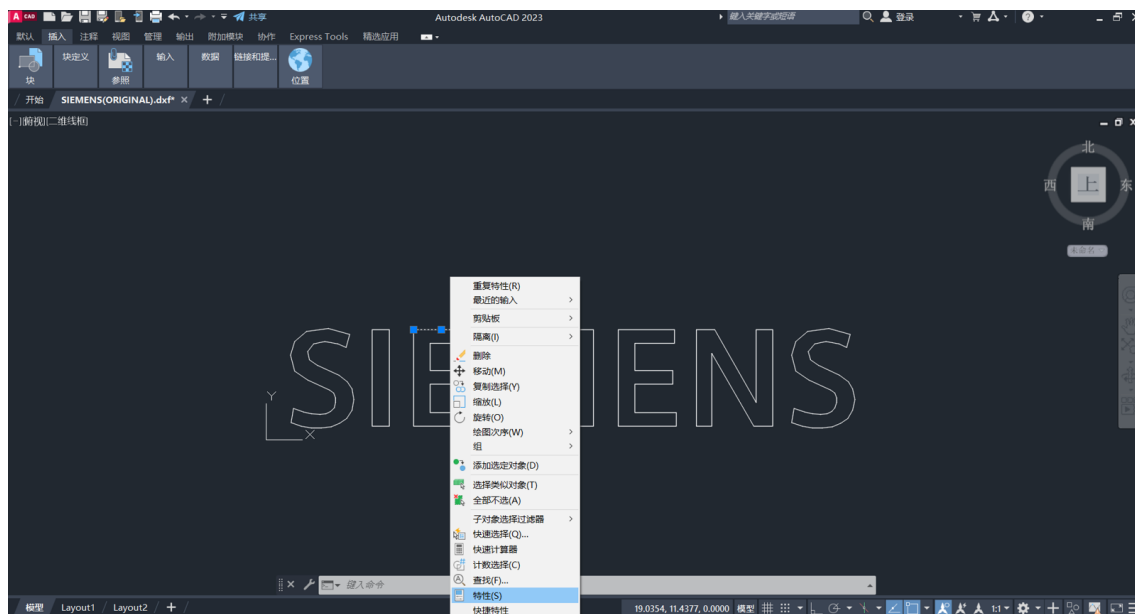
如果坐标的点位和想要的差距过大甚至坐标方向有误，很有可能是基准的原点位置没有设置正确，可以通过在软件中输入快捷指令“UCS”，将十字光标拖放到正确的位置后单击空格键就可以重置基准原点。

7 简单的坐标获取方法

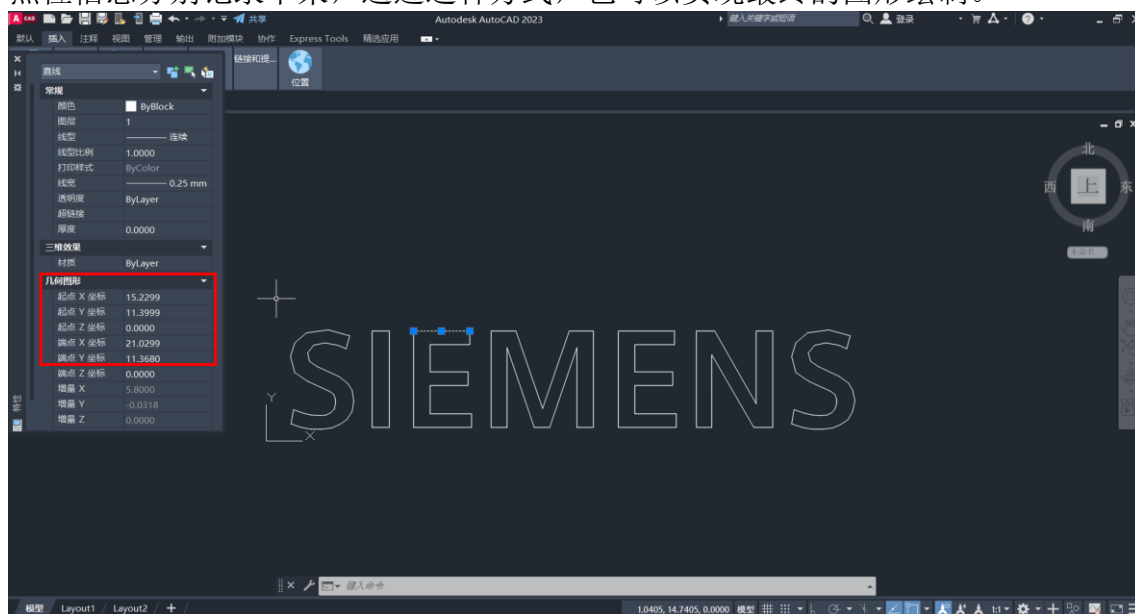


在重新定位确定好原点后，使用光标选择任意一条线段后单击鼠标右键，选择特性。

7 简单的坐标获取方法



选择成功后在软件的左侧便会显示选中线段的起点和终点的坐标，将每条线段的点位信息分别记录下来，通过这种方式，也可以实现最终的图形绘制。



8 执行结果

