**Aerotech DIW printer manual**

Architected Materials Laboratory

April 2024

1. Introduction to the Aerotech Gantry

The gantry system from Aerotech is an AGS15000HL Series Stage. It features a single X-axis guide rail, a primary Y-axis guide rail, a secondary YY-axis guide rail, and four independent Z-axis lift platforms, labeled A, B, C, and D respectively. For detailed technical information on the gantry technology, please refer to the User Guide and the Aerotech official website. This document primarily explains the installation, usage, and control considerations of the Gantry. It also covers methods for customizing it into a DIW (Direct Ink Writing) 3D printer and adding other functionalities.

* 1. Important Usage Consideration

1. Electrical Installation

The stage's protective ground is located on the motor power connector. This ground must be connected to a ground connection (the cables provided by Aerotech will create this connection).

1. Air cooling

It is recommended that air cooling be utilized for the linear motors if duty cycle, speeds, and/or accelerations are high. t is recommended that 20 psi air be available at the motor inlet for adequate cooling.

1. Maximum Load

The maximum load on the upper axis of the system is 40 kg. Additionally, the gantry's movement speeds are limited. The maximum speed for the X-axis is 75 mm/sec with a maximum displacement of 650 mm. For the Y-axis, the speed limit is also 75 mm/sec, with a maximum displacement of 500 mm. The YY-axis functions as a follower axis and typically does not have an independent speed setting. The four Z-axes, labeled A, B, C, and D, have a maximum movement speed of 40 mm/sec and a maximum displacement of 150 mm. **If the gantry exceeds these speed or displacement limits, its operation will halt immediately**. The problematic command will be highlighted, requiring the operator to stop the system, reset, adjust the erroneous command, and restart the operation.

1. Pause and Resume Printing

The Aerotech Gantry supports pausing and resuming printing. Simply press the pause button, and the Gantry will temporarily stop moving. Pressing start again will resume printing from the point where it was paused.

1. Emergency Stop Switch

The emergency stop switch is located on the right side of the gantry. Pressing it will immediately halt the gantry's operation. After an emergency stop, the switch must be reset before the gantry can be used again.

1. Installation

Because the -M1 option (Download from Aerotech.com) was selected for the software, no CD or USB drive will be present. I apologize for any inconvenience this has caused. Below is the documentation for your system:

The Customer Information folder, including drawings, system specifications, and the .MCD file pre-configured to run the system. And the Automation1 Help document available on our website.

[Customer Files 630337.zip](https://urldefense.com/v3/__https:/support.aerotech.com/servicedesk/customershim/secure/attachment/149893/149893_Customer*Files*630337.zip?fromIssue=45165__;Kys!!IBzWLUs!XQ0s27OKBjRHmpi1osHrKVZyTeqjcqh0jpIAUCFTV4IRO9q0H8ElhmbwO5ZtTSISAD-PWazjjhpuMUu0YTM$)

[https://help.aerotech.com/automation1/Content/MainPage.htm](https://urldefense.com/v3/__https:/help.aerotech.com/automation1/Content/MainPage.htm__;!!IBzWLUs!XQ0s27OKBjRHmpi1osHrKVZyTeqjcqh0jpIAUCFTV4IRO9q0H8ElhmbwO5ZtTSISAD-PWazjjhpuN8z_iBw$)

Instructions for installing your software and configuring your PC for best Automation1 performance can be found in the Help documents, here:

[https://help.aerotech.com/automation1/Content/Software-Setup.htm](https://urldefense.com/v3/__https:/help.aerotech.com/automation1/Content/Software-Setup.htm__;!!IBzWLUs!XQ0s27OKBjRHmpi1osHrKVZyTeqjcqh0jpIAUCFTV4IRO9q0H8ElhmbwO5ZtTSISAD-PWazjjhpu_HEt_0g$)

Lastly, you can download your software here:

[https://www.aerotech.com/resources-software-downloads-activation/](https://urldefense.com/v3/__https:/www.aerotech.com/resources-software-downloads-activation/__;!!IBzWLUs!XQ0s27OKBjRHmpi1osHrKVZyTeqjcqh0jpIAUCFTV4IRO9q0H8ElhmbwO5ZtTSISAD-PWazjjhpuqdanKWM$)

To connect the PC to the other electronics, the main action item will be to install the HyperWire card in the PC, then connect the HyperWire card to the first XR3 drive (running six axes). The first XR3 will then daisy-chain into the second XR3 (running only one axis) using the HyperWire cables. Info on installing and configuring the HyperWire card, if necessary, can be found here:

[https://help.aerotech.com/automation1/Content/HyperWire-Card-Overview.htm](https://urldefense.com/v3/__https:/help.aerotech.com/automation1/Content/HyperWire-Card-Overview.htm__;!!IBzWLUs!XWtitD0yvU_UzeKFNm_B62XVckFXiSspum2ZG0Ps86znf2OrvGyzIk42m-WvA5aJbrRxQp80yqJzCoUeaCE$)

A few more items: The zip archive above contains a folder named System Documentation. This folder has .pdf files of the complete system interconnect drawings, including:

Connecting the XR3 to the junction panel (XR3 (SYSTEM INTERCONNECT #1),

The Emergency Stop/Safe Torque Off wiring (SYSTEM INTERCONNECT EFA01213 and the two other EFA01213 drawings), and

Each individual stage (the AGS, PRO, and BMS drawings), including cable management system arrangement, and pinouts for each connector.

If you need any additional schematics for the controller and the common stages, you can find them here: <https://www.aerotech.com/resources-manuals-help-files/>

* 1. Aerotech Gantry Control

1. Aerotech Automation Studio

The Aerotech Gantry is equipped with a control console named Aerotech Automation Studio. The interface is depicted as follows:

图形用户界面, 应用程序, Word

描述已自动生成

The "Command Portal" serves as the area for entering commands; details are provided in section c). Below this, the "Axis Controller" offers the most convenient method for controlling the Gantry. It allows users to Enable/Disable specific axes, set their mode of movement (speed control or displacement control), and adjust their speeds, as well as directly control system movements. The console also features advanced functionalities such as data collection, monitoring of Gantry movements, and visualization display. At the top, there is a toolbar for selecting Aeroscripts (scripts that can be executed by the Gantry), controlling program start/continue, pause, stop, and resetting the Gantry.

For a tutorial video on the Aerotech studio, visit: <https://www.aerotech.com/automation1-studio-overview/>. Additional examples and explanations of the Aerotech Studio can be found on this website: <https://help.aerotech.com/automation1/Content/Using-Automation1.htm#>. Furthermore, examples of Aeroscripts can be located in the 'Example' directory.

1. API Control (Python)

In addition to controlling the printer directly through Automation1 Studio, various APIs provided by Aerotech can be used, including those for C, .NET, and Python programming. According to our setup, all external systems are controlled by Arduino, and communication with the entire system is managed via Python. Therefore, Python is also used to control the Gantry. There are two basic approaches to using Python:

First, **Using Python for Automation1 Studio Tasks**. This approach involves using Python commands to enable axes, home the gantry, run Aeroscripts, and perform other functions available in Automation1 Studio, although the specific operational commands for the Gantry still originate from Aeroscripts. For example, the following commands:

文本

描述已自动生成

It should be noted that many of the functions come from automation1.py, which is an official Python library provided by Aerotech for controlling the Aerotech gantry.

Second, **Direct Control via Python Code**. Python code can also be used to directly control the movements of the gantry. When the printing target is relatively simple (able to be described directly with lines and arcs), Python can be used to directly control both the movement of the Gantry and the air pressure settings, bypassing the need for Automation1 Studio. For example, the following commands:



For more details on the Python API, please refer to the official website:

https://help.aerotech.com/automation1/Content/APIs/Python/References/Controller-Python.htm

Additionally, examples of Python usage can be found in the 'Example' directory.

1. Command Portal

The Command Portal is where specific commands for controlling the Gantry are entered, managing the movement of its various axes. The types of commands accepted include:

• Several RS-274 compatible G-codes and M-codes

• Variables and data types

• Built-in data types (integer, real, axis, handle, string (UTF-8))

• User-defined data types (structures, enumerations)

• Arrays and Programming constructs

• Conditionals: if/elseif/else, switch/case

• Looping: while repeat, for, foreach

• User-defined functions and scoped variables

• User-defined properties

• User-defined libraries

• Built-in Automation1 API

o Math

o Motion

o Analog and Digital I/O

o Files

o Strings

o Transformations

o Custom Callbacks

Among which we mainly use Automation1 functions and Gcode in our application.

1. Printer Customization Plan

To expand future modification options and enhance the potential for upgrades, we have decided not to utilize Aerotech's additional axes (the gantry includes two extra axes, details of which can be found in the User Manual). Instead, all printing functions will be integrated into an Arduino board. This setup will use Python code to simultaneously control gantry motion and printing-related hardware, facilitating various forms of 3D printing.

* 1. Pressure Source for DIW Printing

The DIW is a 3D printing process in which a viscous ink as its raw material is extruded from a syringe with pressure. Usually there are two ways of generating pressure.

1. Air Source

Using compressed gas to generate air pressure is the most common method. In Room 414, the wall-mounted compressed gas source is connected to the building's gas supply, capable of delivering up to 100 Psi of pressure. There is a valve control switch at the output end. (**To ensure laboratory safety, remember to cut off the gas flow after each experiment**) Additionally, a pressure regulator is available to reduce the output pressure. The compressed gas is directed towards two uses: one part is used for cooling the instrument's motor (usually not required), and the other part flows through pressure regulation systems comprising a proportional solenoid valve, solenoid valve, and pressure sensor. These pressure control systems supply syringes to extrude materials, which, combined with the gantry movement, facilitates DIW (Direct Ink Writing) printing.

1. Piston

Using servo motors to drive pistons that extrude material under pressure is a method capable of producing pressures far exceeding 100 Psi. Currently, this printer does not have servo motors and their control systems installed; however, installing and controlling them is relatively simple. It only requires connecting to a power source and using Arduino to precisely control parameters such as speed and position. **It is important to note that after printing is completed, the piston-driven pressure generation method requires time to depressurize, which means that the material will continue to extrude for a period even after printing has finished**. It is advisable to design code that allows the piston to automatically retract to aid in depressurization, or to leave a redundancy area in the STL design process to prevent damage to the printed parts.

* 1. Pressure Regulation System

To achieve air pressure regulation for DIW printing, we have constructed two sets of pressure regulation systems. Each system includes the following components: a proportional solenoid valve (KPI-VP-05-A0-13-V), a solenoid valve (FESTO 196137), a 5V relay, a shared Arduino board (Mega 2560), a MOSFET (IRF630), a pressure sensor (ELVH-015A-HRND-C-NSA4 and SSCDANN150PGAA5), a 5V power source, a 24V power source, a diode, and a resistor (100Ω). The wiring diagram for these systems is as follows:

图示, 示意图

描述已自动生成

1. Proportional solenoid valve

The proportional solenoid valve receives an analog signal ranging from 0-255 (0-5V) from the Arduino and controls the size of the valve opening based on this signal. Our testing has shown that when the input signal is between 0 and 150, the valve remains essentially closed. Between 150 and 255, the valve gradually opens, which may be related to the amplification function of the MOSFET. In the pressure regulation system, we control the opening proportion of the valve through the Proportional, Integral, Derivative (PID) control of air pressure. This allows the system to maintain the desired air pressure value.

1. Solenoid valve

The solenoid valve serves two primary functions. Firstly, it helps the system quickly depressurize after printing is completed to prevent continued extrusion of material, which could affect the printing results. Secondly, during the printing process, if an anomaly is detected, the valve can immediately cut off the connection between the air source and both the syringe and nozzle, facilitating an emergency stop. The working voltage of this solenoid valve is 24V. Therefore, its activation and deactivation are controlled by an Arduino signal, which is relayed through a 5V to 24V relay.

1. Pressure sensors

Given that the maximum air pressure in the laboratory's pneumatic system does not exceed 125 psi, we have selected pressure sensors with a range of 0-150 psi. The primary function of these sensors is to measure the air pressure inside the nozzle and to perform feedback control to achieve and maintain a set pressure. It is important to note that the pressure we use is measured relative to the ambient air pressure (standard atmospheric pressure is approximately 15 psi), which means it is a gauge pressure relative to the atmosphere.

2.3 Communication between the gantry and the pressure regulation system

To facilitate DIW printing, it is necessary to achieve interactive operation between the gantry and the pressure regulation system. This process is implemented through Python code.

1. Gantry and Python

As previously mentioned in section 1.2 b), the gantry's motion control can be achieved via the Python API. However, it is important to note that the operation of the gantry and the execution of Python code occur in parallel. This means that when we use the "run\_program" command to execute an aeroscript on the gantry, the code does not wait for the aeroscript to finish running; instead, it continues executing the remaining commands. Therefore, we need to set a global variable that remains at 1 while the aeroscript is running and changes to 0 once it is complete. This global variable determines the operation of the pressure regulation system. In the aeroscript, this global variable is managed with the initial command "$iglobal[0]=1" and the final command "$iglobal[0]=0". Correspondingly, in the Python code, we have set up the following loop to ensure synchronization between the pressure regulation system and gantry motion:

文本

描述已自动生成

1. Pressure regulation system (Arduino) and Python

Controlling Arduino and its associated hardware via Python is already a well-established approach. In this system, Python sends numerical data to Arduino, where the "system number" specifies the pressure regulation system being controlled. Subsequent numbers control the output of the pressure regulation system. If the input is -1, the solenoid valve will open; if the input is -2, the solenoid valve will close. If the input ranges from 0 to 100, this number (measured in psi) will be set as the target air pressure for PID control. These functionalities are implemented in the Python code through custom functions: close\_valve (system\_number), open\_valve (system\_number), and set\_PID\_pressure (system\_number, pressure).

1. User interface

To facilitate real-time adjustment of the two pressure regulation systems, we have designed a user interface (UI) using the Tkinter package. This UI features two buttons: "toggle valve 1" and "toggle valve 2," which control the opening and closing of the solenoid valves. Additionally, there are two dialogue boxes, "set pressure 1" and "set pressure 2," where users can input numbers between 0 to 100 to adjust the target air pressure in real time. The target air pressure will be outputted in the Python console in real time.

1. Use Example

In this section, we will illustrate how to use the Aerotech printer with an example.

**Step 1**: Import the STL file into Simplify3D software. Choose appropriate printing parameters (such as nozzle size, printing speed, and infill percentage) to generate G-code. For example, here we generated a G-code document named "Triangle\_pair\_tensile.gcode." However, its format does not fully meet the requirements for Aeroscript and needs further modification.

**Step 2**: Place the G-code into the “Gcode\_revision” directory and run the “Gcode\_revision\_v2 script”. This script can perform the following functions:

a) Remove redundant commands generated by Simplify3D, such as controlling the Extruder motor and temperature controls.

b) Add "$iglobal[0]=1" at the beginning of the G-code and "$iglobal[0]=0" at the end to indicate the gantry's operational status.

c) Set the current nozzle position as the coordinate origin.

d) Convert the Z-axis to A, B, C, and D axes.

e) If the distance of a single movement exceeds 10, split the step into multiple steps to prevent triggering a “MaxVelocity Warning”.

As shown in the image below, after running the script, we obtained the modified “Triangle-pair\_tensile\_revised.gcode” document.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**Step 3**: Open Aerotech Automation1 Studio, select "open new file," copy the contents of “Triangle-pair\_tensile\_revised.gcode” into the new file, and save it in the default directory (located at: This PC\Local Disk\ProgramData\Aerotech\Automation1-iSMC\fs\user). As illustrated below, we saved the file as “Triangle\_pai\_tensile.aeroscript”.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**Step 4**: Open “work\_combined\_with\_hardware\_printerUI-UIasmainthread\_final\_version.py” program. Set the initial air pressures to the variables called “desired\_pressure\_1” and “desired\_pressure\_2” (both must be integers). Enter the filename of the previously saved aeroscript in the run\_program() statement within the printing\_process() function, as shown in the following image:

文本

描述已自动生成

**Step 5**: Connect the 5V and 24V power supplies to the Arduino system and turn on the laboratory gas valve.

桌子上有许多电子产品和电脑

描述已自动生成

**Step 6**: Place the printing base. Use the Automation1 Studio Axial Controller to complete the nozzle calibration. The final position of the nozzle is the starting point for printing.

**Step 7**: Run “work\_combined\_with\_hardware\_printerUI-UIasmainthread\_final\_version.py”. This will launch a UI at the same time, where you can control the solenoid valves by clicking “Toggle Valve 1” and "Toggle Valve 2." You can also enter new target air pressure values **(which must be integers)** in the “Set Pressure 1” and “Set Pressure 2” dialogue boxes.

**Step 8**: If an emergency stop is necessary, terminate the code in the Python console and stop the Gantry operation in Automation1 Studio. After the experiment is completed, close the laboratory gas valve, and disconnect the 24V and 5V power supplies that power the Arduino system.

1. Hardware and Software

4.1 Hardware

a) Proportional solenoid valve:

<https://kellypneumatics.com/product/miniature-proportional-valve/?network=g&device=c&keyword=&campaign=1576965577&adgroup=pla-519425395631&gclid=CjwKCAjwv8qkBhAnEiwAkY-ahm7stZjV1okNO5cHzmOdkCIpvXhZ8q-M-3PNOkViJfvUWaZ17FIjyBoCAxsQAvD_BwE>

1. ELVH-015A-HRND-C-NSA4 analog pressure sensor:

<https://www.cdiweb.com/products/detail/elvh015ahrndcnsa4-all-sensors/682138/>

1. SSCDANN150PGAA5 I2C pressure sensor:

digikey.com/en/products/detail/honeywell-sensing-and-productivity-solutions/SSCDANN150PGAA5/2416090

1. Festo 196137 solenoid valve:

<https://www.sp-spareparts.com/en/p/mhe2-m1h-3-2g-qs-4-festo?curr=usd&utm_source=adwords&utm_medium=ppc&utm_term=&utm_campaign=PMax+%7C++Festo+%7C+US&hsa_cam=19750235214&hsa_mt=&hsa_ver=3&hsa_src=x&hsa_ad=&hsa_net=adwords&hsa_tgt=&hsa_acc=6705635644&hsa_grp=&hsa_kw=&gad=1&gclid=CjwKCAjwv8qkBhAnEiwAkY-ahreYTaSSIh6r1elyNCXxjkHijSgjBpQ7-CAg2zrRPVZgrN17gz9r9xoCxWoQAvD_BwE>

4.2 Software

1. **Double pressure regulation system Arduino**

**Arduino\20240408final\_2PID\_speed\_optimization.ino**

1. **Double pressure regulation system with UI Python**

**work\_combined\_with\_hardware\_printerUI-UIasmainthread\_final\_version.py**

1. Douvle pressure regulation system without UI Python

work\_double\_channel\_final\_version.py

1. Single pressure regulation system based on analog sensor Arduino

Arduino\20230912\_final\_PID.ino

1. Single pressure regulation system Python

work\_single\_channel\_final\_version.py

1. PID control based on I2C sensor Arduino

20240328\_N2A4sensor\_PID\_only.ino

1. automation1 package

automation1\_0912.py

1. **Gcode revision**

**Gcode\_revision\Gcode\_revision\_v2.py**

1. Outlook
   1. FDM printing module

图形用户界面

描述已自动生成

Makergear: (<https://makergear.com/collections/parts/products/extruder-motor?variant=21668418500>)

5.2 Aerosol Jetting module

Optomec Print Engine (AJ 5X System, Optomec Inc., Albuquerque, New Mexico, USA) with a pneumatic atomizer (https://optomec.com/ajprintengine/)

图形用户界面, 文本

描述已自动生成

5.3 Photonic curing module

Xenon flash lamp (model RC-847, Xenon Corp., USA) (https://www.excelitas.com/product-category/flashlamps)

图片包含 文本

描述已自动生成图片包含 应用程序

描述已自动生成

5.4 other possible hardware

a) laser calibration system

b) heated plate

c) heated nozzle

d) UV curing light

e) rotating nozzle

f) electro-magnet and magnetic shield

g) core-shell nozzle

h) cameras