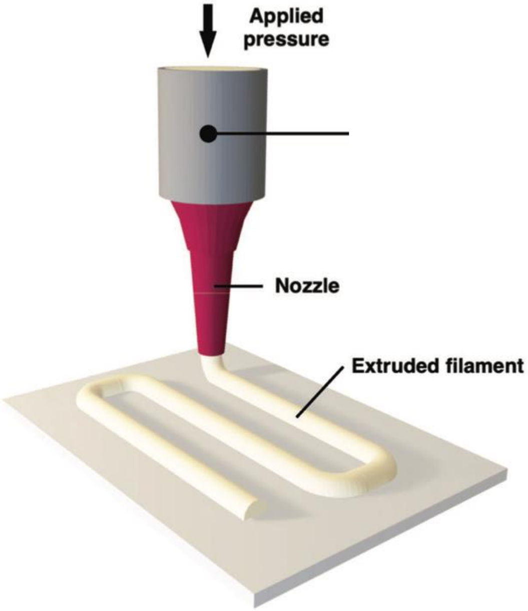
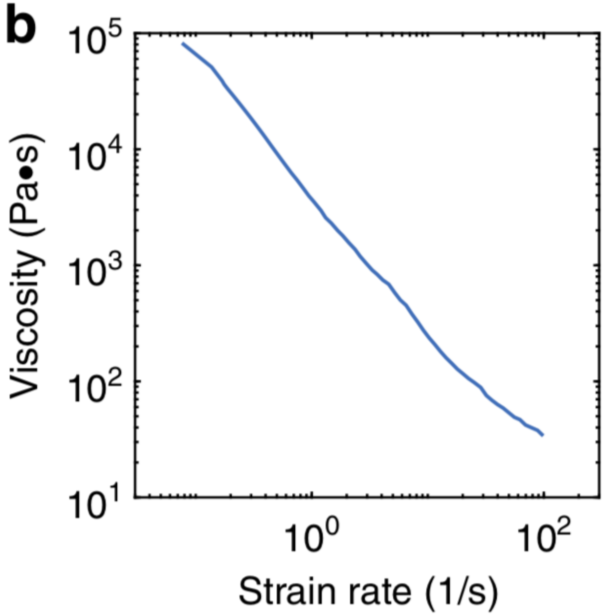
Aerotech DIW printer customization

Yucong Hua, Zebang Zhang

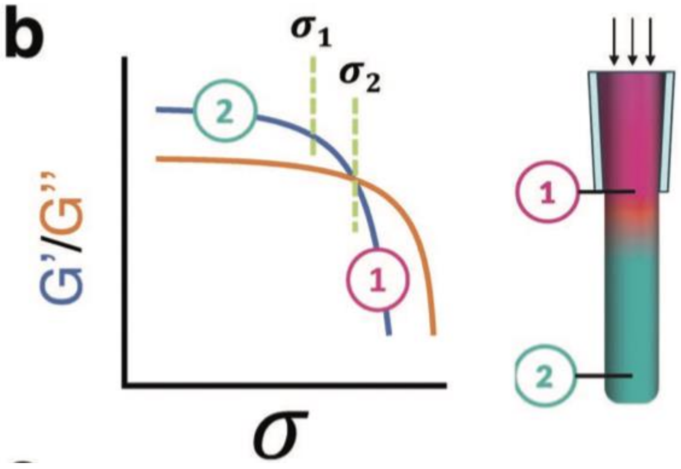
2023/09/26

Direct Ink Writing(DIW) 3D printing





Shear-thinning: viscosity decreases with increasing shear rate



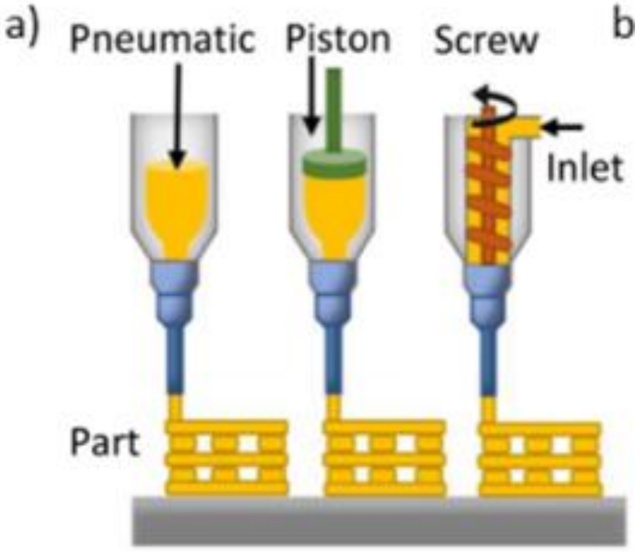
Viscoelastic behavior: retain shape fidelity

M.A.S.R. Saadi et al. Adv. Mater. 2022, 34, 2108855; Y. Jiang et al. Nature Communications, 2019;10:128

Viscoelastic, shear-thinning inks are methodically pushed through a nozzle to construct various 3D forms

Inside the nozzle: shear stress is above yield stress, ink yields and flows
Outside the nozzle: material transforms to a viscoelastic solid

Direct Ink Writing(DIW) 3D printing



Surjadi et al. 2019, Adv. Eng. Mater.

Method to generate pressure:

Piston; Screw; Air source.

Process parameters:

Extrusion pressure;

Printing Speed;

Nozzle diameter;

Layer height;

Line spacing coefficient.

Ink parameters:

Viscosity;

Post-printing curing methods;

Chemistry/crosslinking time/fillers.

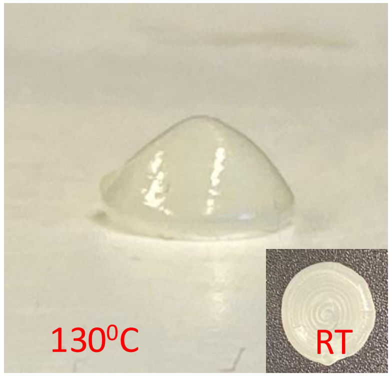
Applications:

- **Structural:** complex architectures, porous structures, functional composites, biomimetics;
- **Soft robotics:** soft machines, self-folding structures, autonomous robots, actuators;
- **Electronics:** capacitor, micro-battery, wearable electronics, sensors, LEDs, transistors;
- **Biomedical:** artificial organs, bioimplants, microvascular networks, scaffolds, drug delivery;
- **Food:** chocolate, cookie, cheese, fruits, meat, vegetables, edible hydrogels.

Direct Ink Writing(DIW) 3D printing

Inks used: Polymers including Silicone rubber, Liquid Crystal Elastomer, Hydrogel, Epoxy, Magnetic soft materials, Conductive polymer, Ceramics, Glass, Cement, and so on.

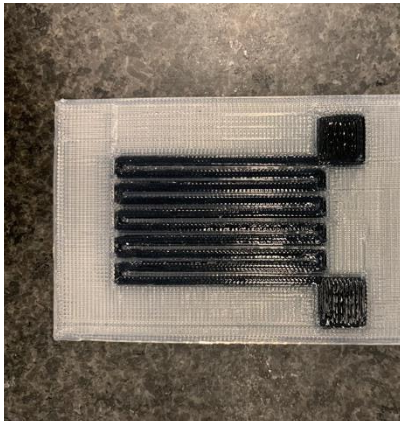
Magnetic soft materials



Liquid crystal elastomers

Hydrogels

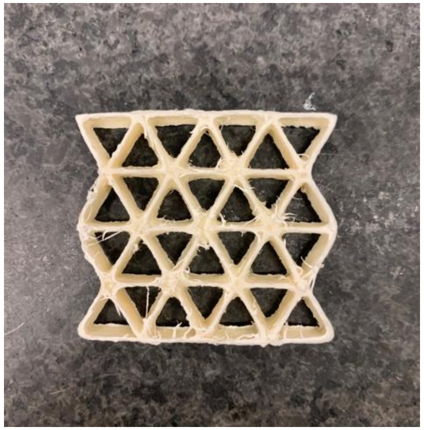
Epoxy



Conductive polymer

Eggshell powder

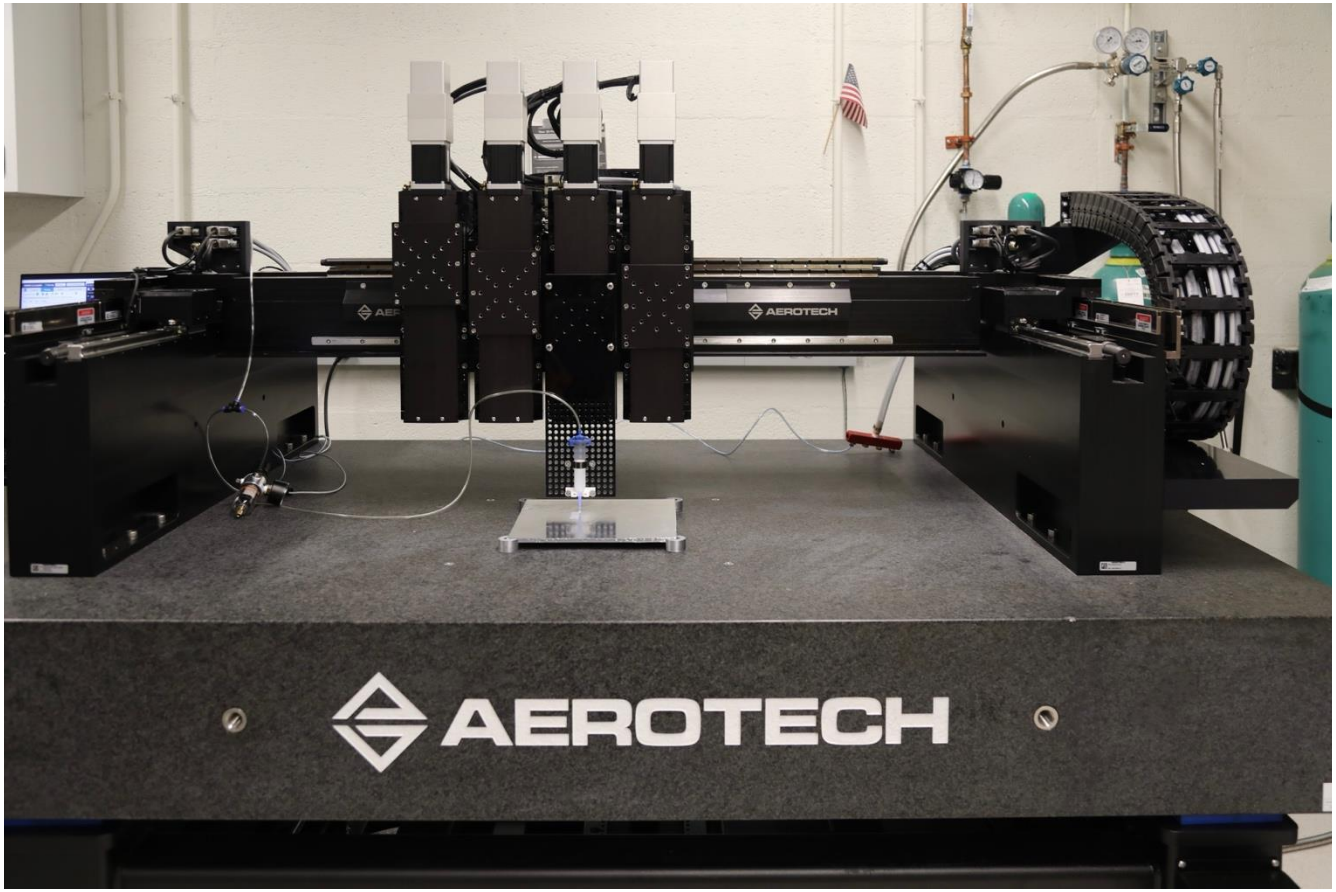
Flour and beef tallow



Sticky rice and cotton fiber

Pro: low-cost, simple, with ability to combine different materials and introduce multifunctionality in a single processing

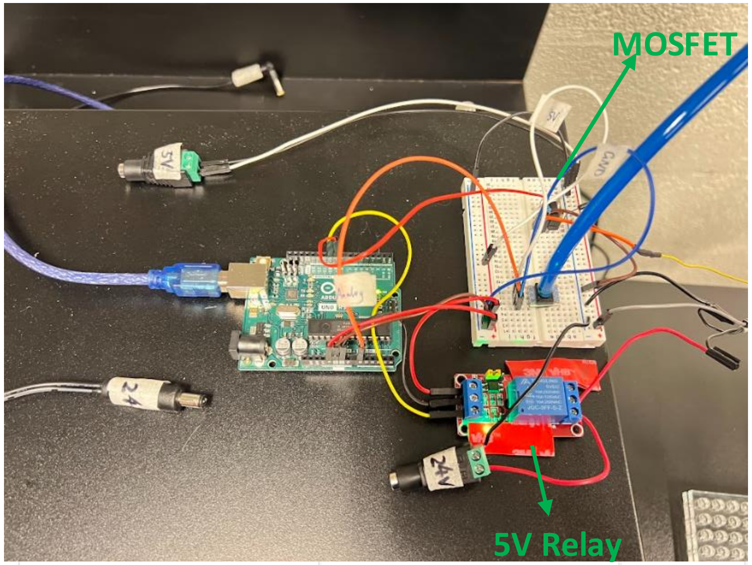
Aerotech Gantry

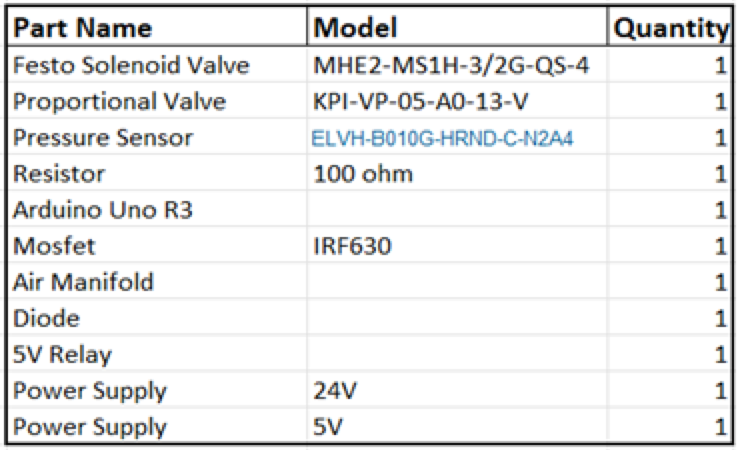


A powerful Gantry, high precision, huge platform

Various modules can be added according to needs

Aerotech DIW printer customization: Component



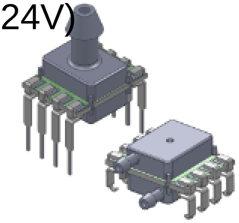


Part Name	Model	Quantity
Festo Solenoid Valve	MHE2-MS1H-3/2G-QS-4	1
Proportional Valve	KPI-VP-05-A0-13-V	1
Pressure Sensor	ELVH-B010G-HRND-C-N2A4	1
Resistor	100 ohm	1
Arduino Uno R3		1
Mosfet	IRF630	1
Air Manifold		1
Diode		1
5V Relay		1
Power Supply	24V	1
Power Supply	5V	1

Air solenoid valve (24V)

Arduino UNO Rev3

5V Relay



Air pressure Sensor

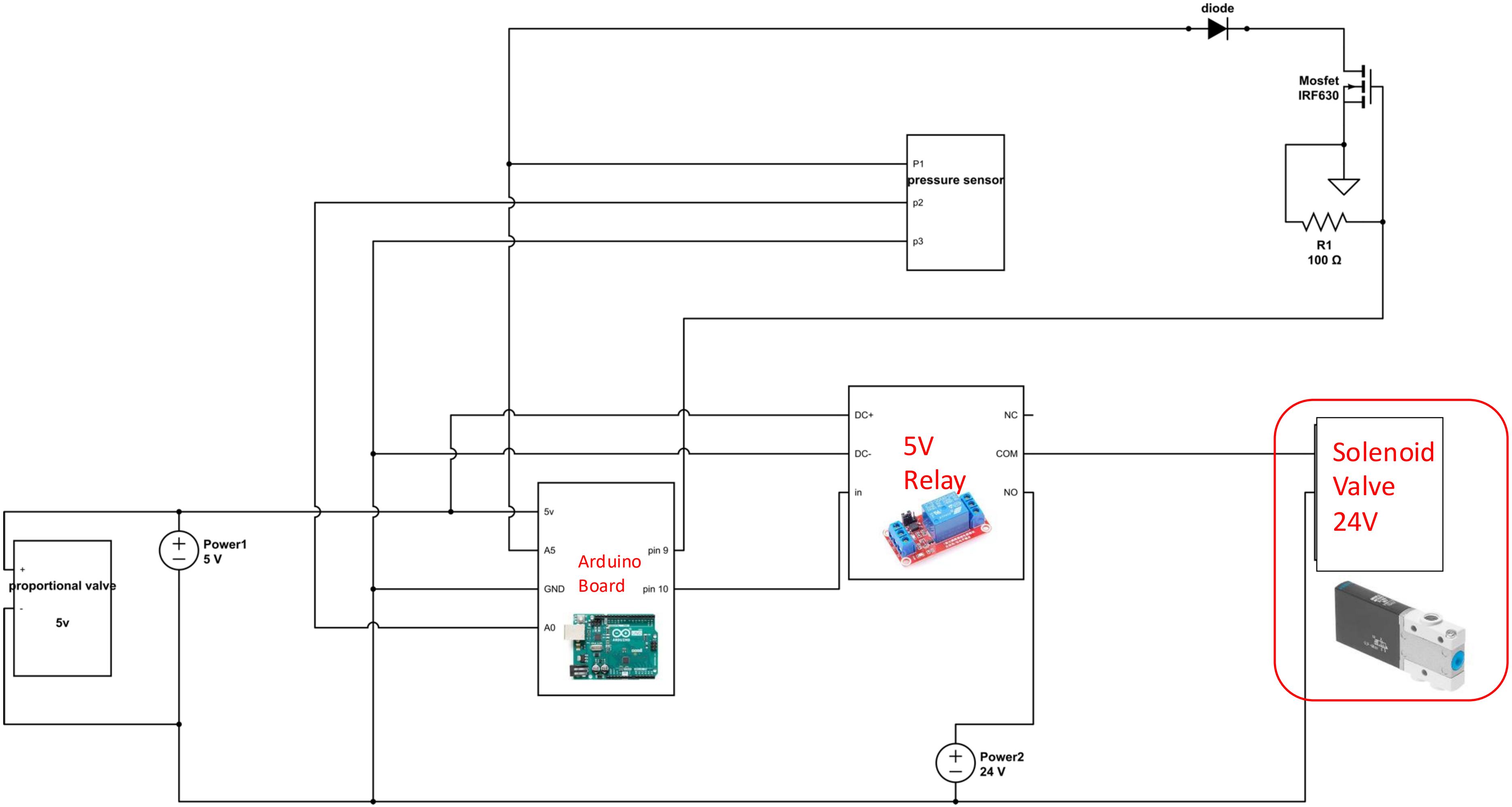
100 ohm Resistor

MOSFET IRF630

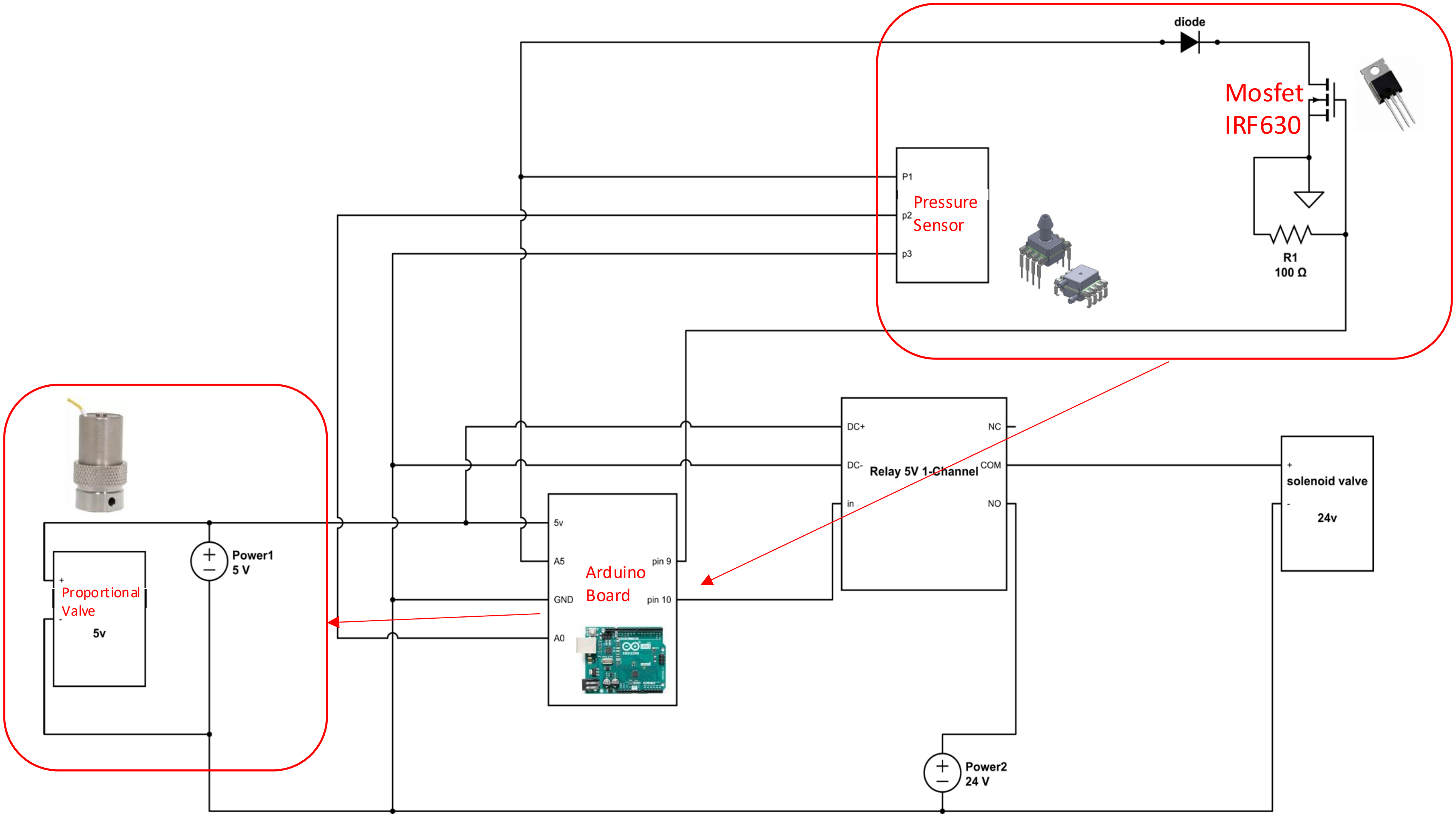
Power Adapter 24V / 5V

Air Manifold

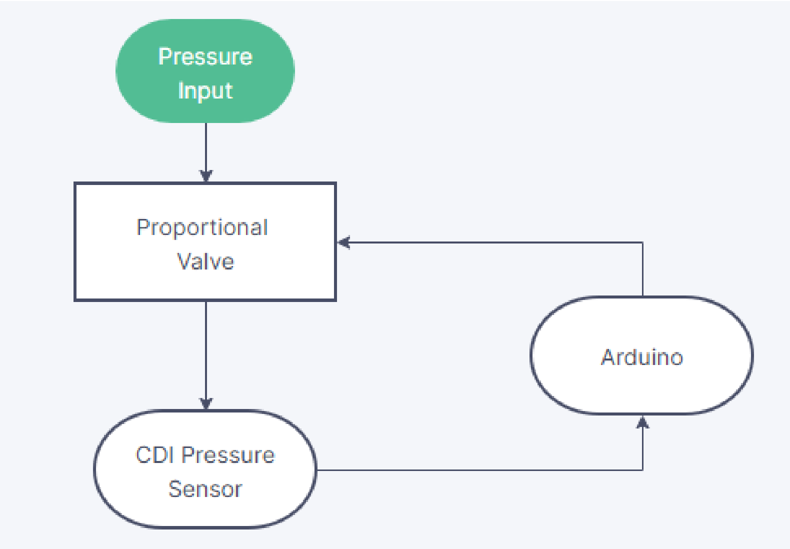
Aerotech DIW printer customization: circuit



Aerotech DIW printer customization: circuit



Aerotech DIW printer customization: control

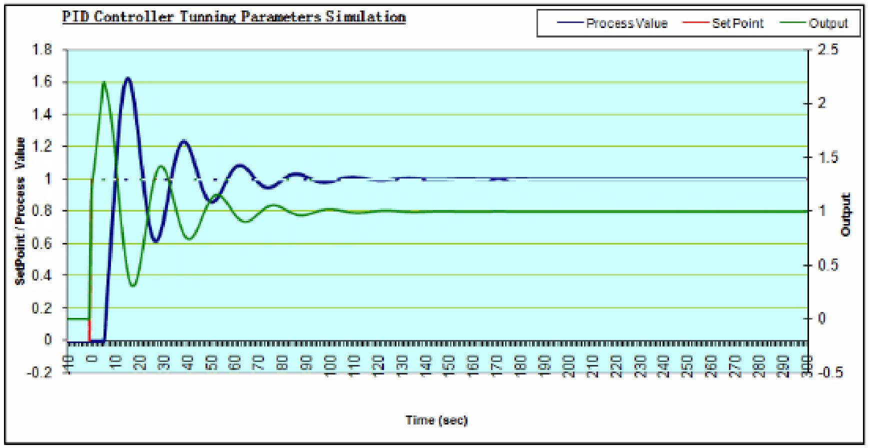


Pressure Input

Proportional Valve

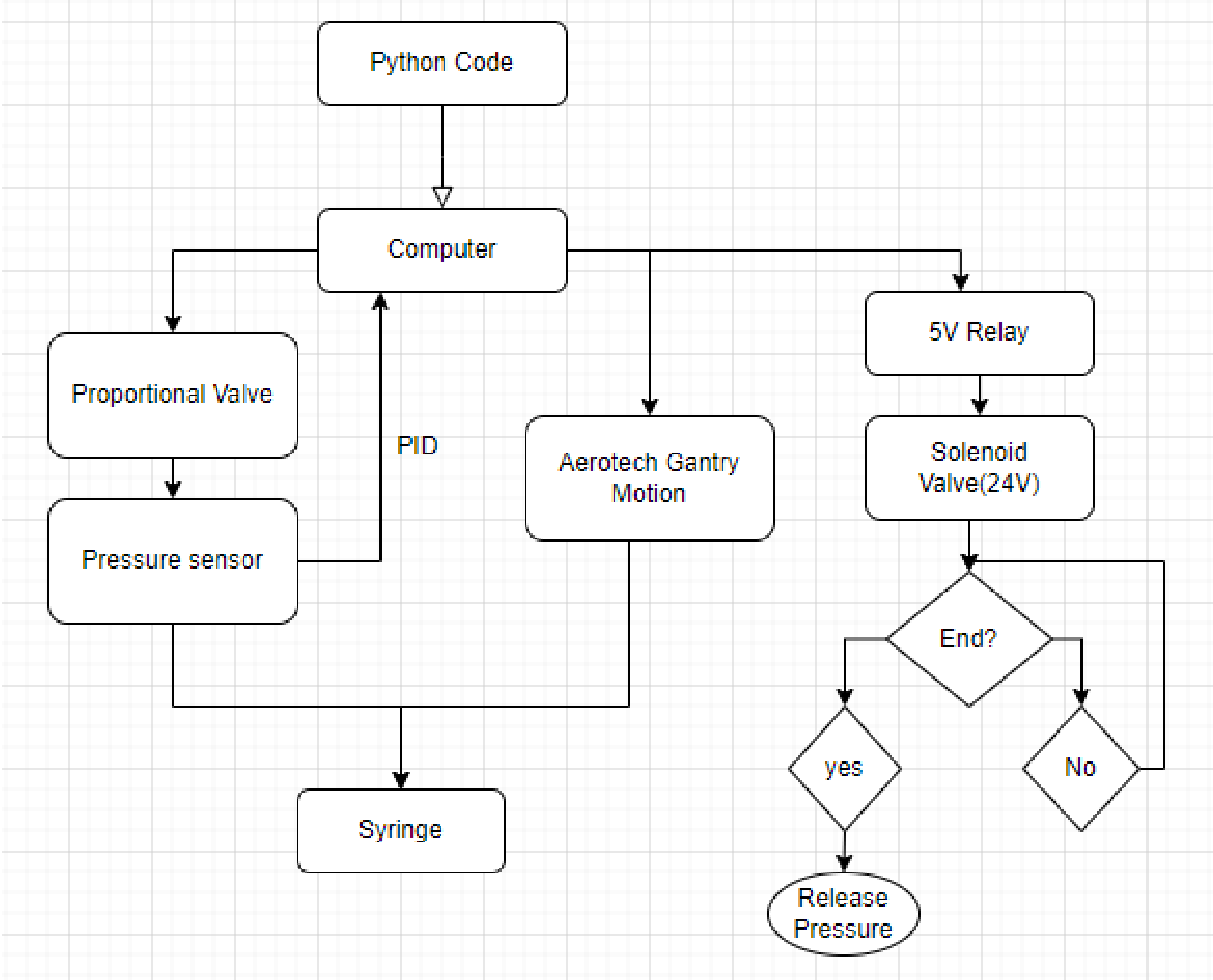
Arduino

CDI Pressure Sensor

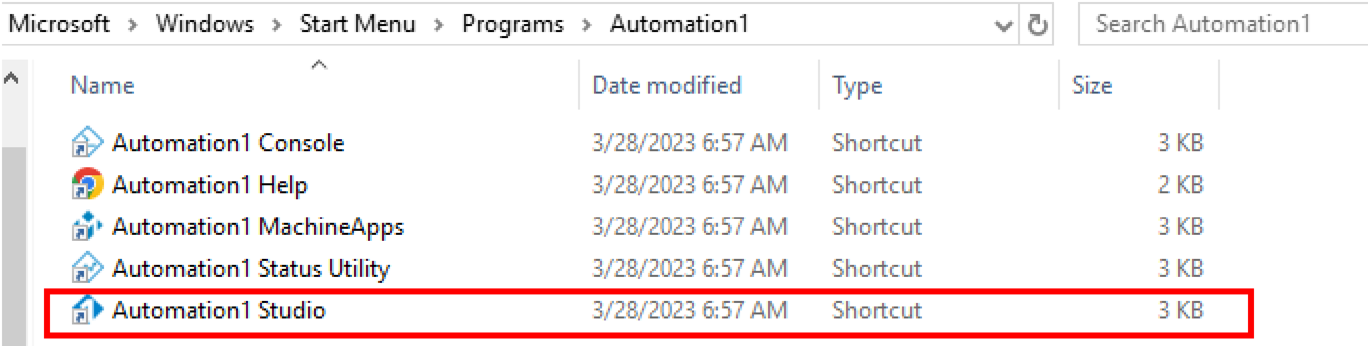


- MOSFET as a Switch
- Proportional Valve Control
- Feedback Loop
- Real-time Adjustments
- Goal Point vs. Measured Value
- Error Minimization

Aerotech DIW printer customization: hardware flow chart



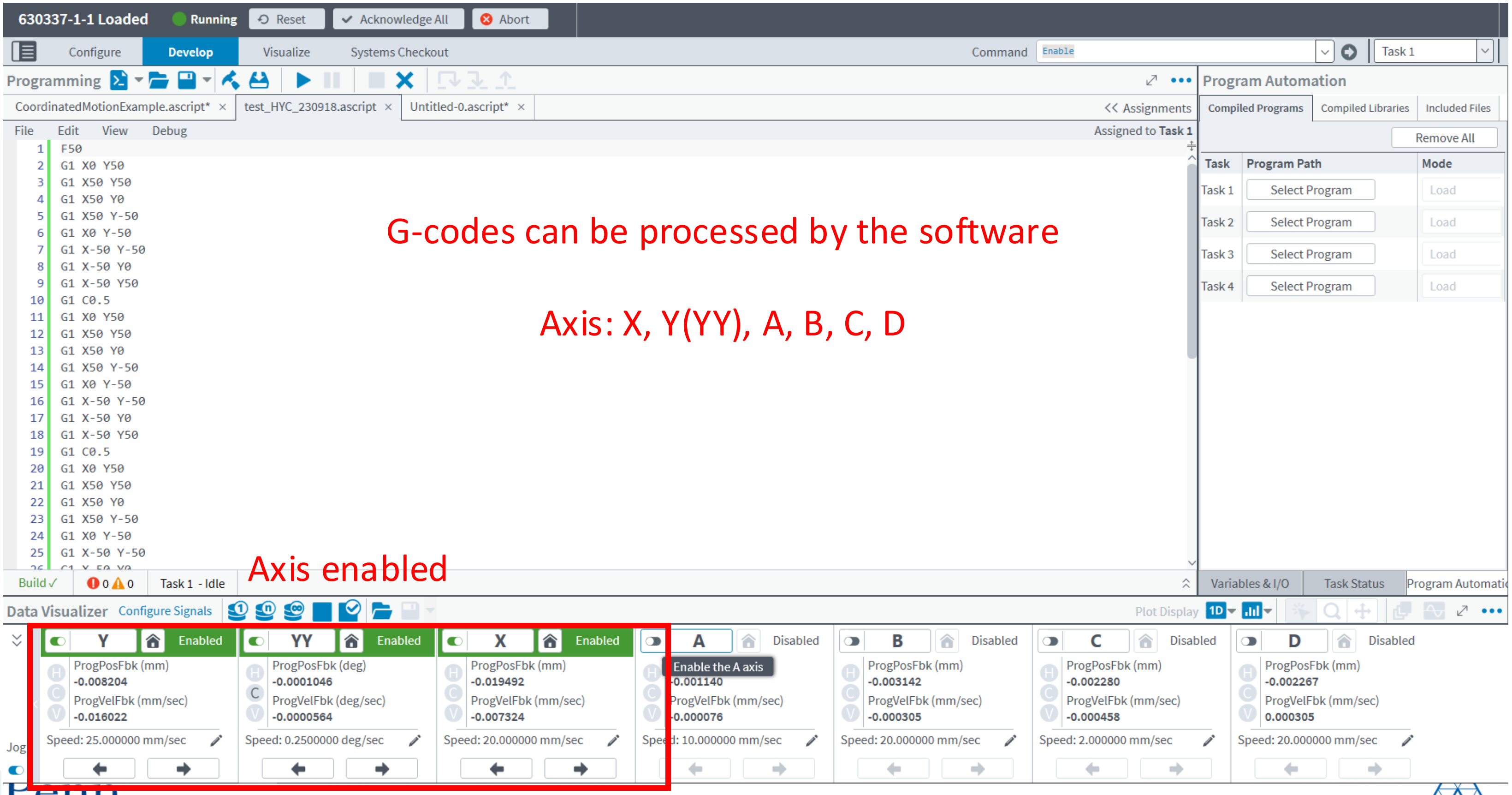
Aerotech DIW printer customization: control



Microsoft › Windows › Start Menu › Programs › Automation1

Search Automation1

Name	Date modified	Type	Size
Automation1 Console	3/28/2023 6:57 AM	Shortcut	3 KB
Automation1 Help	3/28/2023 6:57 AM	Shortcut	2 KB
Automation1 MachineApps	3/28/2023 6:57 AM	Shortcut	3 KB
Automation1 Status Utility	3/28/2023 6:57 AM	Shortcut	3 KB
Automation1 Studio	3/28/2023 6:57 AM	Shortcut	3 KB



```
F50
G1 X0 Y50
G1 X50 Y50
G1 X50 Y0
G1 X50 Y-50
G1 X0 Y-50
G1 X-50 Y-50
G1 X-50 Y0
G1 X-50 Y50
G1 C0.5
G1 X0 Y50
G1 X50 Y50
G1 X50 Y0
G1 X50 Y-50
G1 X0 Y-50
G1 X-50 Y-50
G1 X-50 Y0
G1 X-50 Y50
G1 C0.5
G1 X0 Y50
G1 X50 Y50
G1 X50 Y0
G1 X50 Y-50
G1 X0 Y-50
G1 X-50 Y-50
```

Task	Program Path	Mode
Task 1	Select Program	Load
Task 2	Select Program	Load
Task 3	Select Program	Load
Task 4	Select Program	Load

Axis	State	ProgPosFbk	ProgVelFbk	Speed
Y	Enabled	ProgPosFbk (mm) -0.008204	ProgVelFbk (mm/sec) -0.016022	Speed: 25.000000 mm/sec
YY	Enabled	ProgPosFbk (deg) -0.0001046	ProgVelFbk (deg/sec) -0.0000564	Speed: 0.2500000 deg/sec
X	Enabled	ProgPosFbk (mm) -0.019492	ProgVelFbk (mm/sec) -0.007324	Speed: 20.000000 mm/sec
A	Disabled	0.001140	ProgVelFbk (mm/sec) 0.000076	Speed: 10.000000 mm/sec
B	Disabled	ProgPosFbk (mm) -0.003142	ProgVelFbk (mm/sec) -0.000305	Speed: 20.000000 mm/sec
C	Disabled	ProgPosFbk (mm) -0.002280	ProgVelFbk (mm/sec) -0.000458	Speed: 2.000000 mm/sec
D	Disabled	ProgPosFbk (mm) -0.002267	ProgVelFbk (mm/sec) 0.000305	Speed: 20.000000 mm/sec

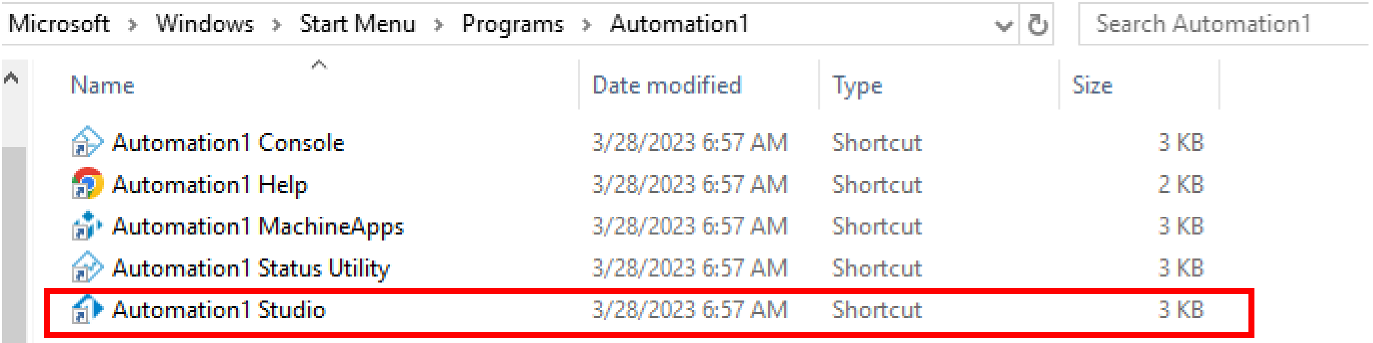
G-codes can be processed by the software

Axis: X, Y(YY), A, B, C, D

Axis enabled

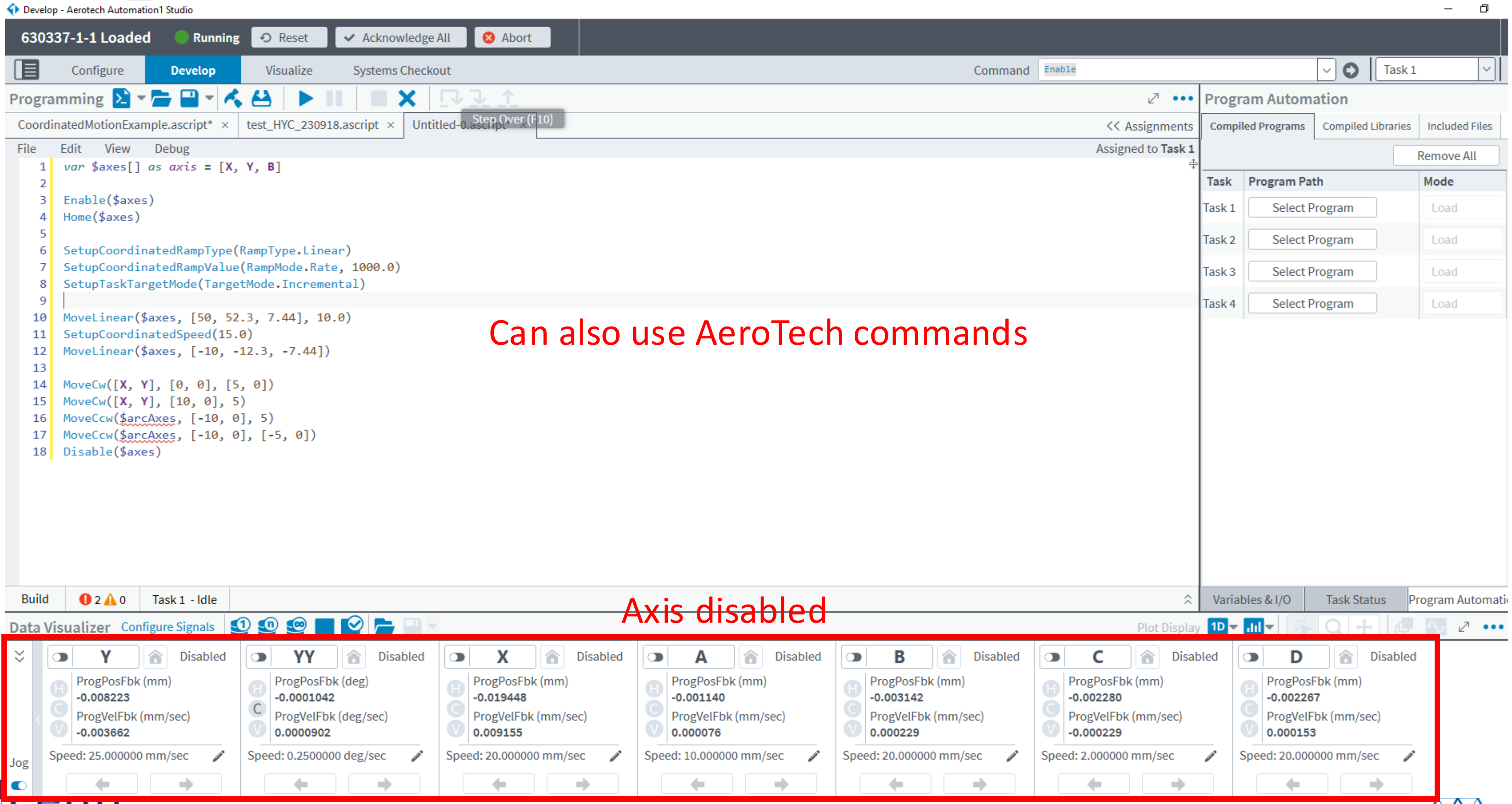
Can directly control the motion using the Left and Right arrows below each axis

Aerotech DIW printer customization: control



Microsoft › Windows › Start Menu › Programs › Automation1

Name	Date modified	Type	Size
Automation1 Console	3/28/2023 6:57 AM	Shortcut	3 KB
Automation1 Help	3/28/2023 6:57 AM	Shortcut	2 KB
Automation1 MachineApps	3/28/2023 6:57 AM	Shortcut	3 KB
Automation1 Status Utility	3/28/2023 6:57 AM	Shortcut	3 KB
Automation1 Studio	3/28/2023 6:57 AM	Shortcut	3 KB



Develop - Aerotech Automation1 Studio

630337-1-1 Loaded

Command: Enable

```
var $axes[] as axis = [X, Y, B]

Enable($axes)
Home($axes)

SetupCoordinatedRampType(RampType.Linear)
SetupCoordinatedRampValue(RampMode.Rate, 1000.0)
SetupTaskTargetMode(TargetMode.Incremental)

MoveLinear($axes, [50, 52.3, 7.44], 10.0)
SetupCoordinatedSpeed(15.0)
MoveLinear($axes, [-10, -12.3, -7.44])

MoveCw([X, Y], [0, 0], [5, 0])
MoveCw([X, Y], [10, 0], 5)
MoveCcw($arcAxes, [-10, 0], 5)
MoveCcw($arcAxes, [-10, 0], [-5, 0])
Disable($axes)
```

Task	Program Path	Mode
Task 1	Select Program	Load
Task 2	Select Program	Load
Task 3	Select Program	Load
Task 4	Select Program	Load

Can also use AeroTech commands

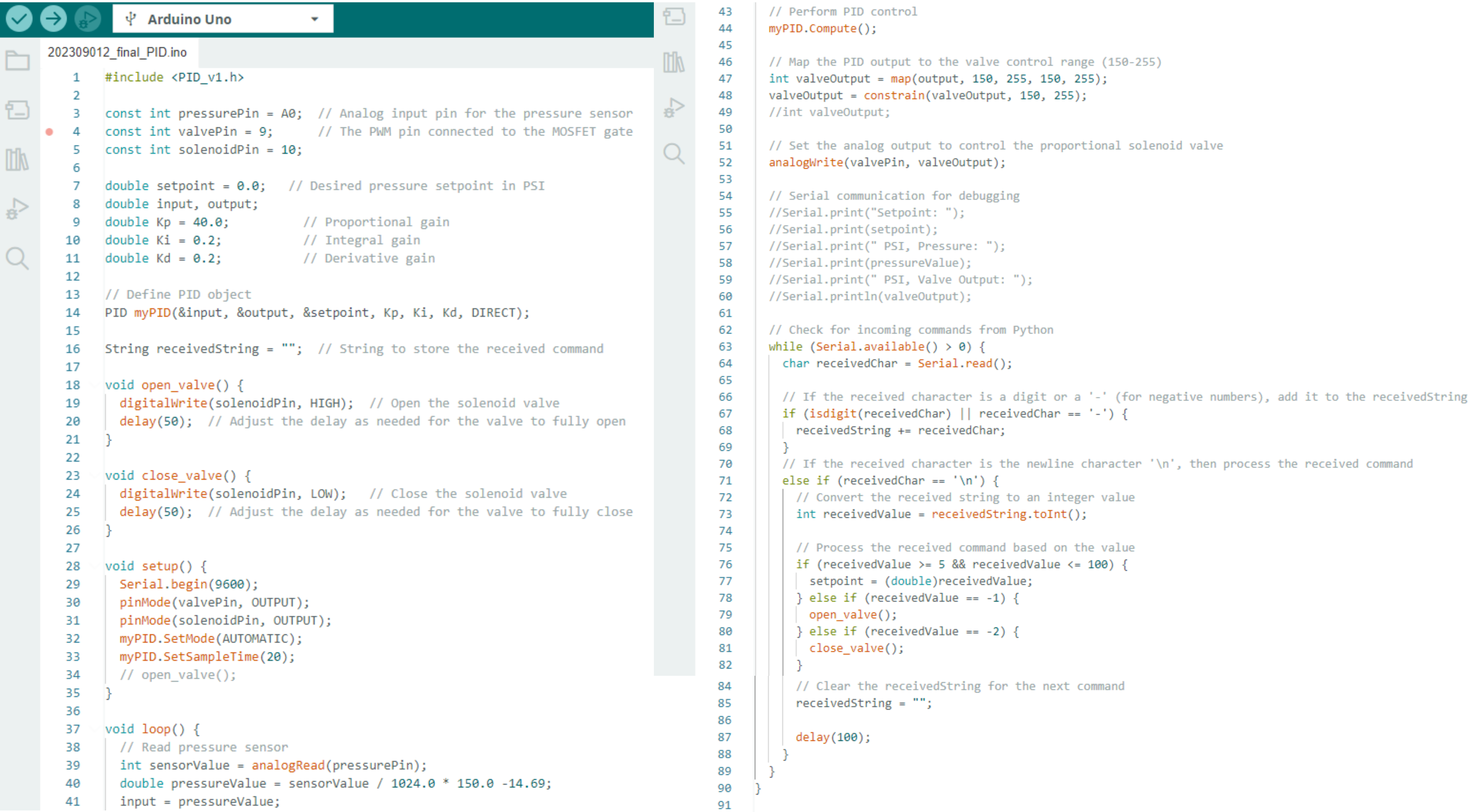
Build | 2 0 | Task 1 - Idle

Axis disabled

Y	YY	X	A	B	C	D
Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
ProgPosFbk (mm)	ProgPosFbk (deg)	ProgPosFbk (mm)	ProgPosFbk (mm)	ProgPosFbk (mm)	ProgPosFbk (mm)	ProgPosFbk (mm)
-0.008223	-0.0001042	-0.019448	-0.001140	-0.003142	-0.002280	-0.002267
ProgVelFbk (mm/sec)	ProgVelFbk (deg/sec)	ProgVelFbk (mm/sec)	ProgVelFbk (mm/sec)	ProgVelFbk (mm/sec)	ProgVelFbk (mm/sec)	ProgVelFbk (mm/sec)
-0.003662	0.0000902	0.009155	0.000076	0.000229	-0.000229	0.000153
Speed: 25.000000 mm/sec	Speed: 0.2500000 deg/sec	Speed: 20.000000 mm/sec	Speed: 10.000000 mm/sec	Speed: 20.000000 mm/sec	Speed: 2.000000 mm/sec	Speed: 20.000000 mm/sec

Aerotech DIW printer customization: control

For pneumatic system: Python to Arduino



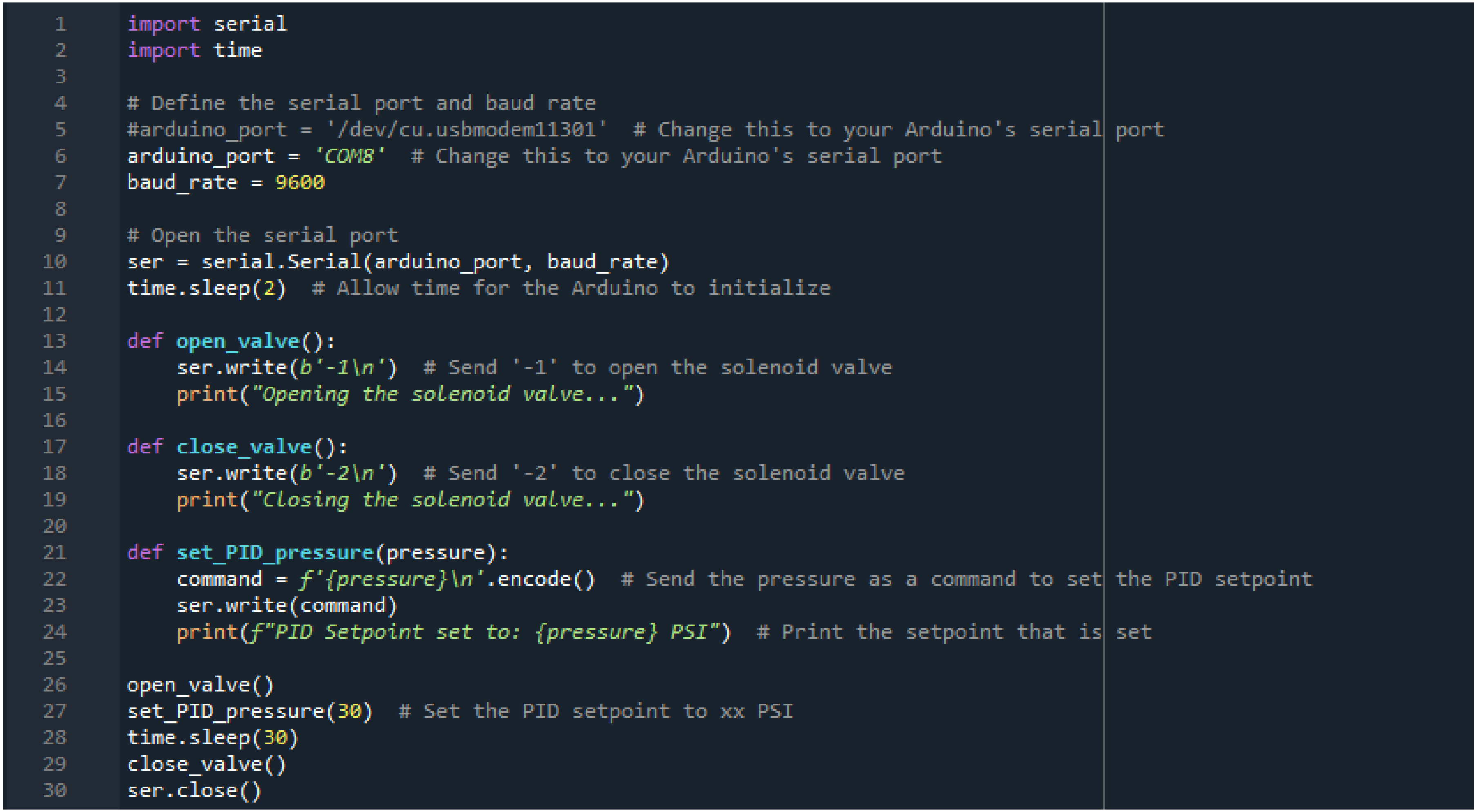
```
#include <PID_v1.h>

const int pressurePin = A0;  // Analog input pin for the pressure sensor
const int valvePin = 9;      // The PWM pin connected to the MOSFET gate
const int solenoidPin = 10;

double setpoint = 0.0;   // Desired pressure setpoint in PSI
double input, output;
double Kp = 40.0;          // Proportional gain
double Ki = 0.2;           // Integral gain
double Kd = 0.2;           // Derivative gain

// Define PID object
PID myPID(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);

String receivedString = "";  // String to store the received command

void open_valve() {
  digitalWrite(solenoidPin, HIGH);  // Open the solenoid valve
  delay(50);  // Adjust the delay as needed for the valve to fully open
}

void close_valve() {
  digitalWrite(solenoidPin, LOW);   // Close the solenoid valve
  delay(50);  // Adjust the delay as needed for the valve to fully close
}

void setup() {
  Serial.begin(9600);
  pinMode(valvePin, OUTPUT);
  pinMode(solenoidPin, OUTPUT);
  myPID.SetMode(AUTOMATIC);
  myPID.SetSampleTime(20);
  // open_valve();
}

void loop() {
  // Read pressure sensor
  int sensorValue = analogRead(pressurePin);
  double pressureValue = sensorValue / 1024.0 * 150.0 -14.69;
  input = pressureValue;
```

```
  // Perform PID control
  myPID.Compute();

  // Map the PID output to the valve control range (150-255)
  int valveOutput = map(output, 150, 255, 150, 255);
  valveOutput = constrain(valveOutput, 150, 255);
  //int valveOutput;

  // Set the analog output to control the proportional solenoid valve
  analogWrite(valvePin, valveOutput);

  // Serial communication for debugging
  //Serial.print("Setpoint: ");
  //Serial.print(setpoint);
  //Serial.print(" PSI, Pressure: ");
  //Serial.print(pressureValue);
  //Serial.print(" PSI, Valve Output: ");
  //Serial.println(valveOutput);

  // Check for incoming commands from Python
  while (Serial.available() > 0) {
    char receivedChar = Serial.read();

    // If the received character is a digit or a '-' (for negative numbers), add it to the receivedString
    if (isdigit(receivedChar) || receivedChar == '-') {
      receivedString += receivedChar;
    }
    // If the received character is the newline character '\n', then process the received command
    else if (receivedChar == '\n') {
      // Convert the received string to an integer value
      int receivedValue = receivedString.toInt();

      // Process the received command based on the value
      if (receivedValue >= 5 && receivedValue <= 100) {
        setpoint = (double)receivedValue;
      } else if (receivedValue == -1) {
        open_valve();
      } else if (receivedValue == -2) {
        close_valve();
      }
      // Clear the receivedString for the next command
      receivedString = "";

      delay(100);
    }
  }
}

```

Arduino part

Aerotech DIW printer customization: control

For pneumatic system: Python to Arduino

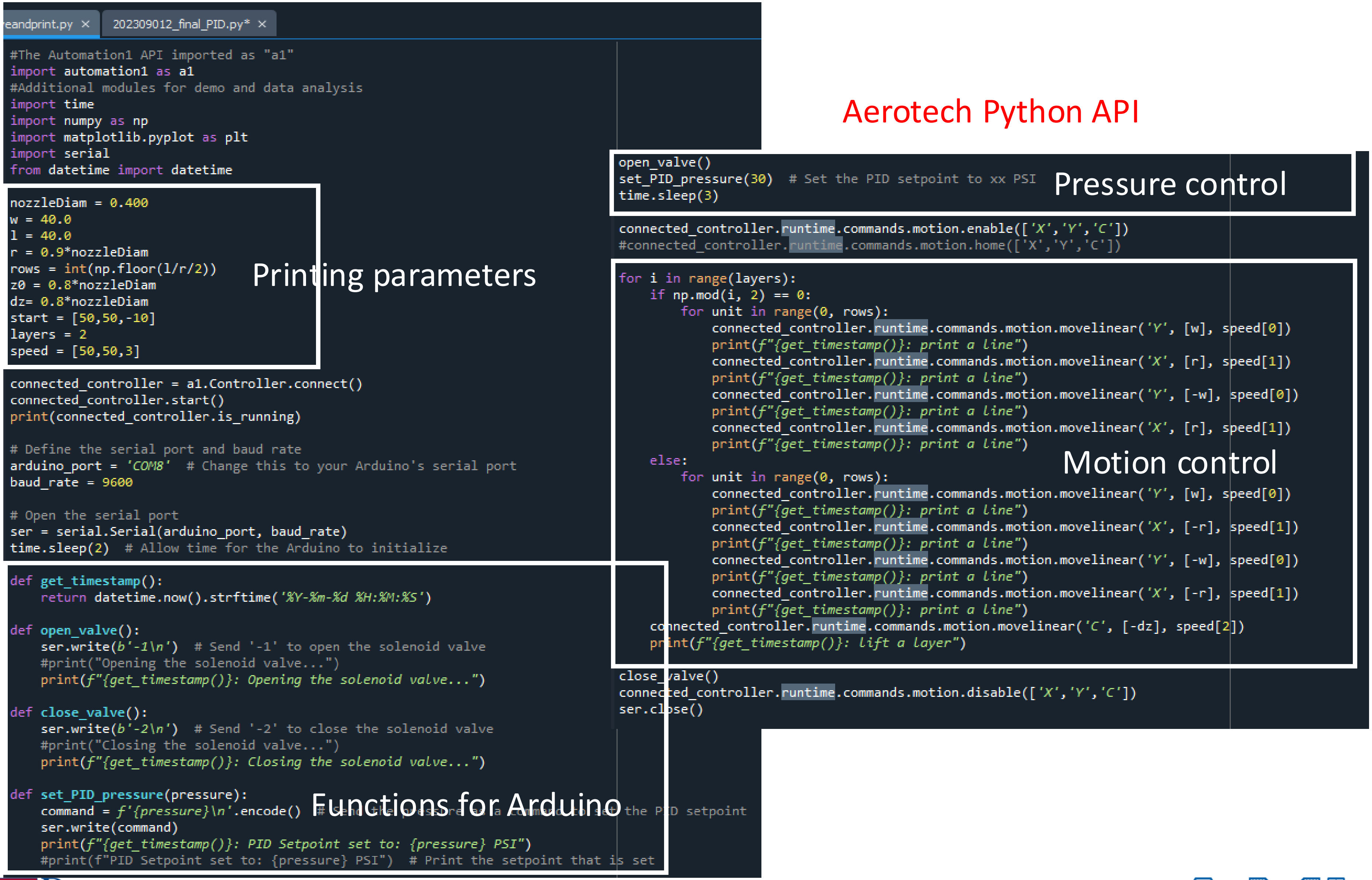


```python
import serial
import time

# Define the serial port and baud rate
#arduino_port = '/dev/cu.usbmodem11301'  # Change this to your Arduino's serial port
arduino_port = 'COM8'  # Change this to your Arduino's serial port
baud_rate = 9600

# Open the serial port
ser = serial.Serial(arduino_port, baud_rate)
time.sleep(2)  # Allow time for the Arduino to initialize

def open_valve():
    ser.write(b'-1\n')  # Send '-1' to open the solenoid valve
    print("Opening the solenoid valve...")

def close_valve():
    ser.write(b'-2\n')  # Send '-2' to close the solenoid valve
    print("Closing the solenoid valve...")

def set_PID_pressure(pressure):
    command = f'{pressure}\n'.encode()  # Send the pressure as a command to set the PID setpoint
    ser.write(command)
    print(f"PID Setpoint set to: {pressure} PSI")  # Print the setpoint that is set

open_valve()
set_PID_pressure(30)  # Set the PID setpoint to xx PSI
time.sleep(30)
close_valve()
ser.close()
```

Python part

Aerotech DIW printer customization: control

Aerotech Python API



```python
#The Automation1 API imported as "a1"
import automation1 as a1
#Additional modules for demo and data analysis
import time
import numpy as np
import matplotlib.pyplot as plt
import serial
from datetime import datetime
```

Printing parameters

```python
nozzleDiam = 0.400
w = 40.0
l = 40.0
r = 0.9*nozzleDiam
rows = int(np.floor(l/r/2))
z0 = 0.8*nozzleDiam
dz= 0.8*nozzleDiam
start = [50,50,-10]
layers = 2
speed = [50,50,3]
```

```python
connected_controller = a1.Controller.connect()
connected_controller.start()
print(connected_controller.is_running)

# Define the serial port and baud rate
arduino_port = 'COM8'  # Change this to your Arduino's serial port
baud_rate = 9600

# Open the serial port
ser = serial.Serial(arduino_port, baud_rate)
time.sleep(2)  # Allow time for the Arduino to initialize
```

Functions for Arduino

```python
def get_timestamp():
    return datetime.now().strftime('%Y-%m-%d %H:%M:%S')

def open_valve():
    ser.write(b'-1\n')  # Send '-1' to open the solenoid valve
    #print("Opening the solenoid valve...")
    print(f"{get_timestamp()}: Opening the solenoid valve...")

def close_valve():
    ser.write(b'-2\n')  # Send '-2' to close the solenoid valve
    #print("Closing the solenoid valve...")
    print(f"{get_timestamp()}: Closing the solenoid valve...")

def set_PID_pressure(pressure):
    command = f'{pressure}\n'.encode()  # Send the pressure as a command to set the PID setpoint
    ser.write(command)
    print(f"{get_timestamp()}: PID Setpoint set to: {pressure} PSI")
    #print(f"PID Setpoint set to: {pressure} PSI")  # Print the setpoint that is set
```

Pressure control

```python
open_valve()
set_PID_pressure(30)  # Set the PID setpoint to xx PSI
time.sleep(3)
```

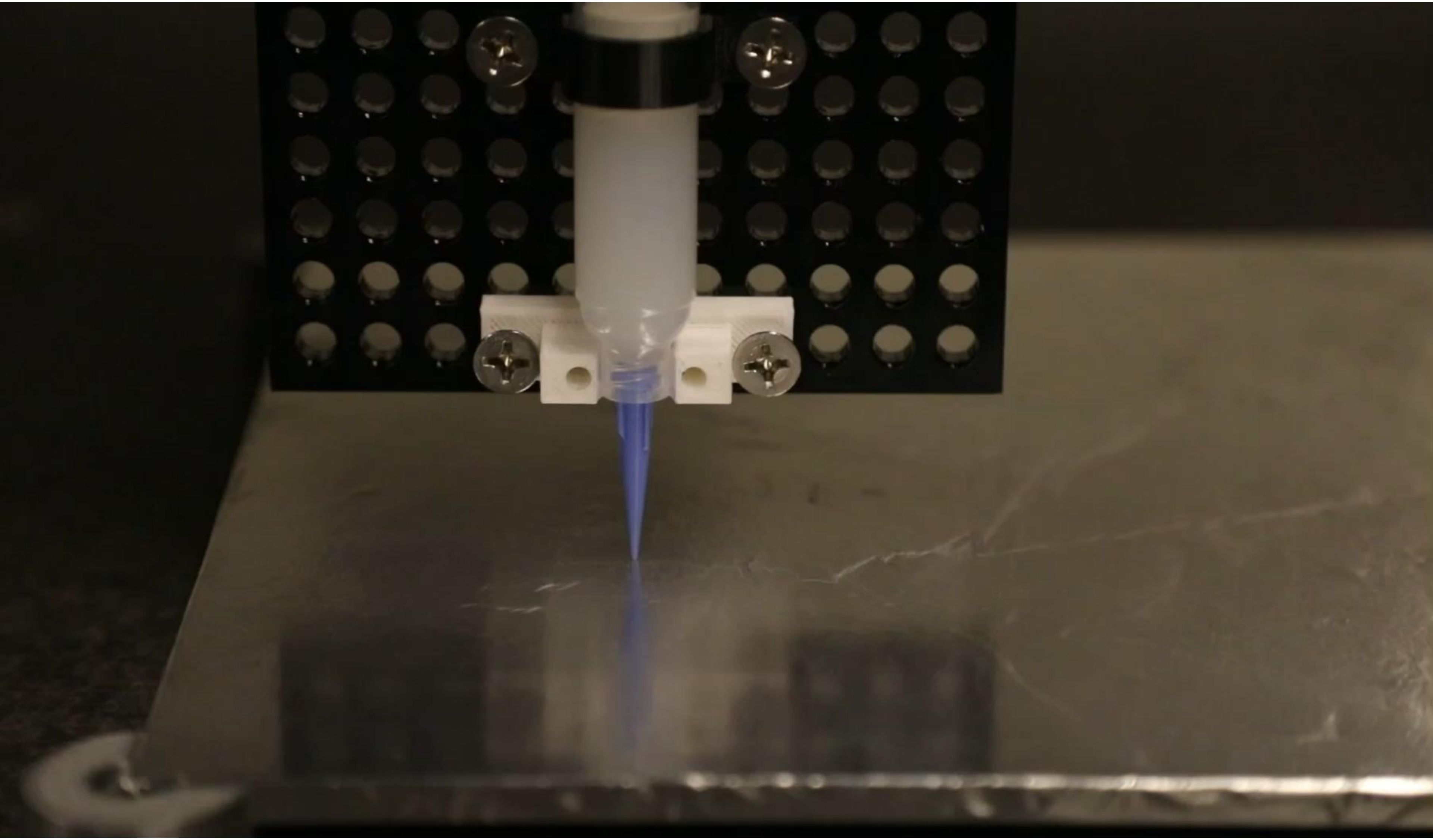
```python
connected_controller.runtime.commands.motion.enable(['X','Y','C'])
#connected_controller.runtime.commands.motion.home(['X','Y','C'])
```

Motion control

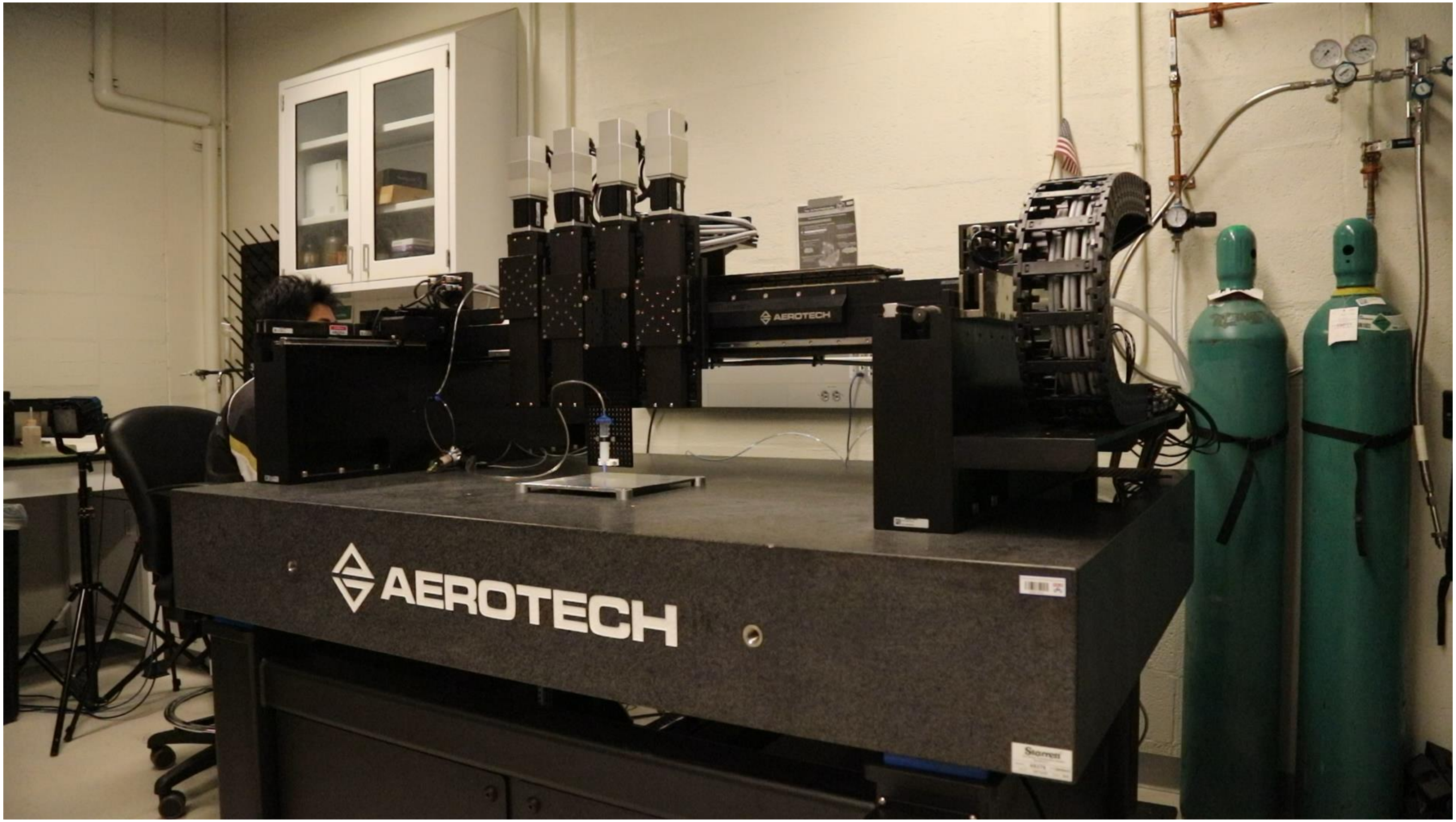
```python
for i in range(layers):
    if np.mod(i, 2) == 0:
        for unit in range(0, rows):
            connected_controller.runtime.commands.motion.movelinear('Y', [w], speed[0])
            print(f"{get_timestamp()}: print a line")
            connected_controller.runtime.commands.motion.movelinear('X', [r], speed[1])
            print(f"{get_timestamp()}: print a line")
            connected_controller.runtime.commands.motion.movelinear('Y', [-w], speed[0])
            print(f"{get_timestamp()}: print a line")
            connected_controller.runtime.commands.motion.movelinear('X', [r], speed[1])
            print(f"{get_timestamp()}: print a line")
    else:
        for unit in range(0, rows):
            connected_controller.runtime.commands.motion.movelinear('Y', [w], speed[0])
            print(f"{get_timestamp()}: print a line")
            connected_controller.runtime.commands.motion.movelinear('X', [-r], speed[1])
            print(f"{get_timestamp()}: print a line")
            connected_controller.runtime.commands.motion.movelinear('Y', [-w], speed[0])
            print(f"{get_timestamp()}: print a line")
            connected_controller.runtime.commands.motion.movelinear('X', [-r], speed[1])
            print(f"{get_timestamp()}: print a line")
    connected_controller.runtime.commands.motion.movelinear('C', [-dz], speed[2])
    print(f"{get_timestamp()}: lift a layer")
```

```python
close_valve()
connected_controller.runtime.commands.motion.disable(['X','Y','C'])
ser.close()
```

Aerotech DIW printer customization: control



Aerotech DIW printer customization: control



Aerotech DIW printer customization: ways of using

Regular structure

Write Python code to control the gantry traverse the whole structure

Combine with pressure Control Python code

Calibrate, set proper pressure, Start printing

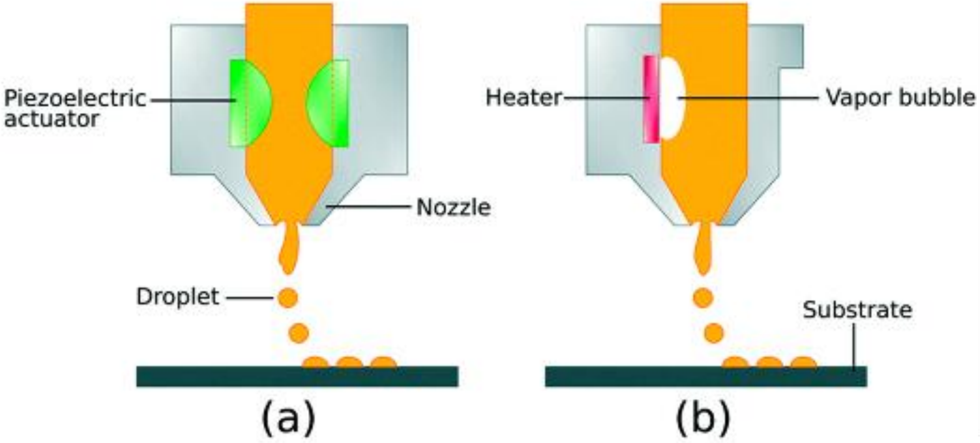
Complex structure

Using slicer to generate Python code for complex structure

Combine with pressure Control Python code

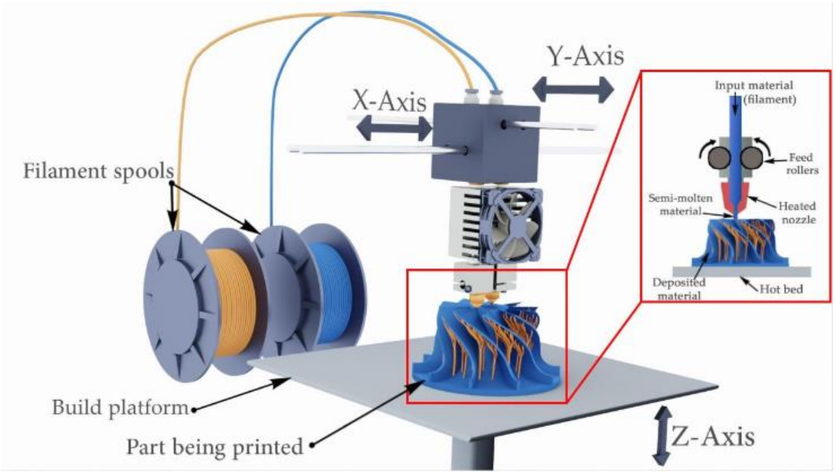
Calibrate, set proper pressure, Start printing

Aerotech Gantry: other possible applications



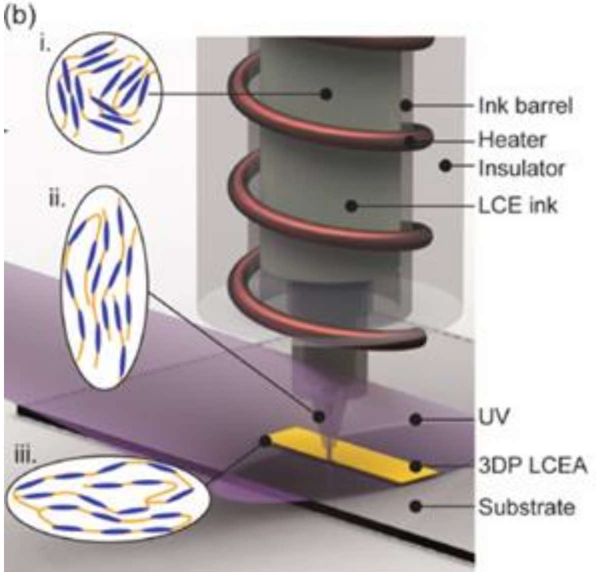
Inkjet 3D printing

Mahmood M A. Compounds, 2021, 1(3): 94-115.



FFF 3D printing

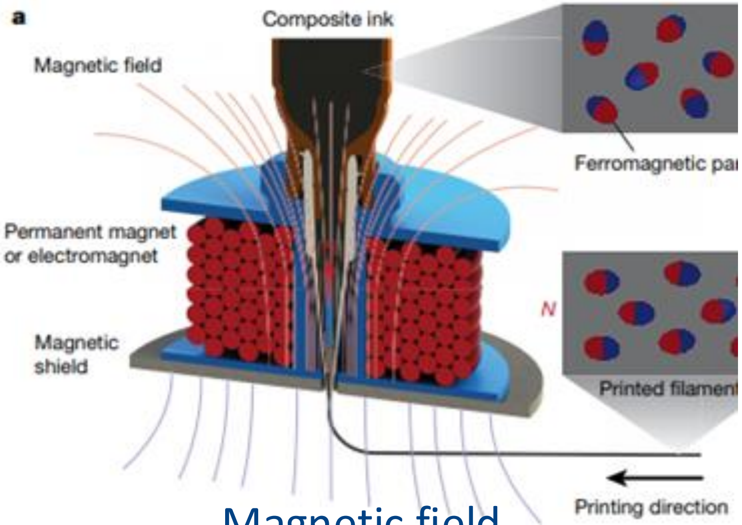
Elkaseer A, Schneider S, Applied Sciences, 2020, 10(8): 2899.



LCE 3D printing with UV light source and Heater

A. Kotikian et al. Adv. Mater. 2018, 30, 1706164

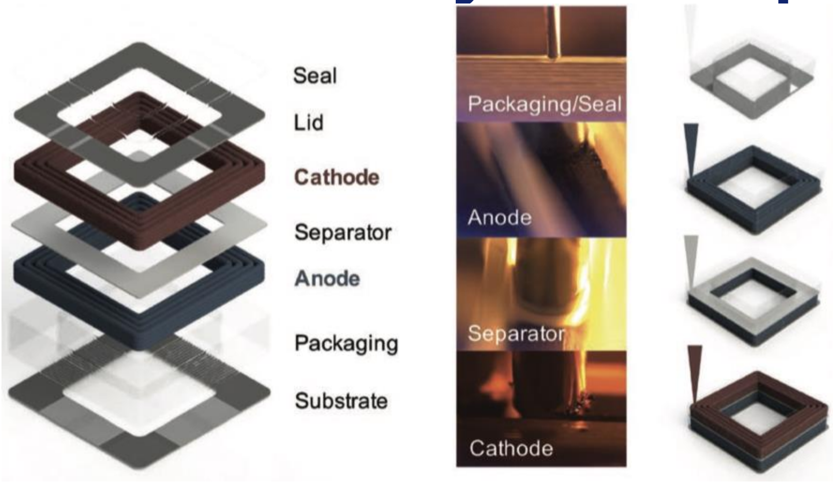
Fused deposition modelling



Magnetic field

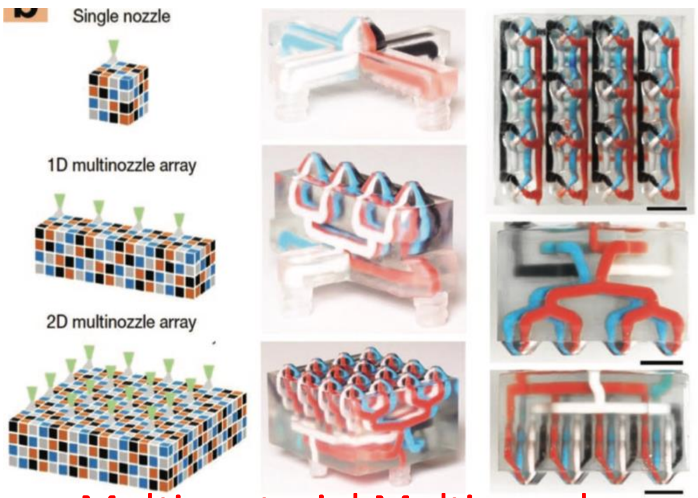
Y. Kim et al., Nature 2018, 558, 274

Aerotech Gantry: other possible applications



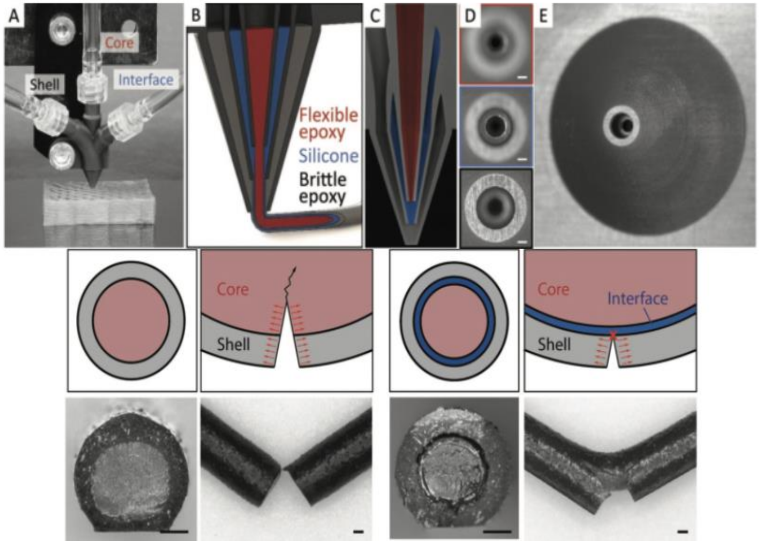
Sequential/Switching nozzles for a Li-ion battery

T.S. Wei et al. Adv. Mater. 2018, 30, 1703027



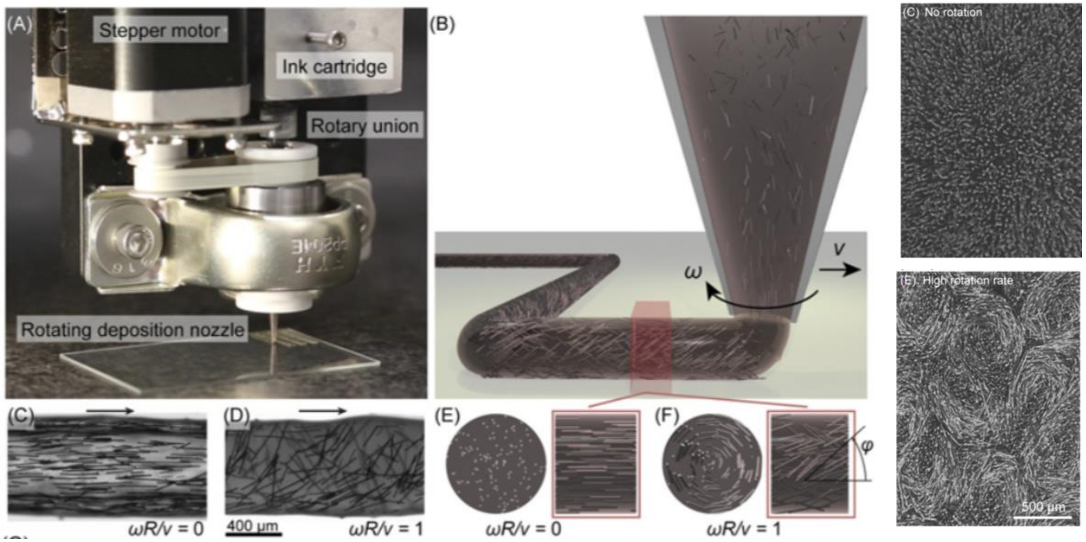
Multi-material Multi-nozzle

M. A. Skylar-Scott et al. Nature 2019, 575, 330



Core shell nozzle for architected materials

J. Mueller et al. Adv. Mater. 2018, 30, 1705001.



Rotational nozzle for damage tolerant composites

J. R. Raney et al. Proc. Natl. Acad. Sci. USA 2018, 115, 1198

Aerotech DIW printer customization: resources

- Use the Automation1 Python API in a Jupyter Notebook
https://www.aerotech.com/use-the-automation1-python-api-in-a-jupyter-notebook/

- Structure of a Program
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/1_Structure-of-a-Program.htm

- G-Code (RS-274) Support
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/9_G-Code-RS-274-Support.htm

- Expressions
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/5_Expressions.htm#Operators-and-Instrinsic-Functions

- Functions
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/7_Functions.htm#User-Defined_Functions

- Comments
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/2_Comments.htm

- Data Types and Variables
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/4_Data-Types-and-Variables.htm#User-Defined_Variables

- Preprocessor
http://help.aerotech.com/automation1/Content/Guide-AeroScript-Programming-Language-Reference/3_Preprocessor.htm

- AeroScript Overview
https://www.aerotech.com/aeroscript-overview/

Thank you!