## LbL Deposition System

Programming Guide

### Main\_Settings.hpp

* File for declaring global constant variables. This includes the main settings for how the sequences will run.

### Main\_Main.cpp

* The main file that will be ran. This contains the default Arduino functions:

setup() and loop().

* + setup()
    - Initializes all components by running their respective setup functions.
  + loop()
    - Loops through for a predetermined number of layers, defined in Main\_Settings.hpp as: const int NUMBER\_OF\_LAYERS
    - The loop sequence is as follows:

Loop 1: positionAtA → 1

Execute the following:

Move to position A, dispenseVolume(), motorLoop(), then write positionAtA from 0 → 1

Loop 2: positionAtA → 0

Move to position B, dispenseVolume(), motorLoop(), then write positionAtA from 1 → 0

Then repeat

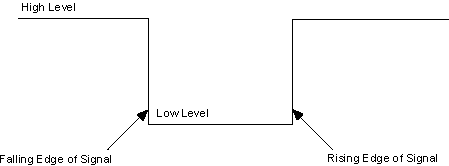
* + - * The alternation of the boolean positionAtA b from true to false (or 1 to 0) changes the argument of the moveToPosition()function from PositionA to Position B

### Main\_BLDC.cpp

* This contains the functions used for controlling the Brushless DC motor.
* Libraries Used: <Servo.h>
  + motorSetup()
    - The setup function for the BLDC motor.
      * Call Location: setup().
  + motorLoop()
    - This is the looping function for the motor.
    - The functions ramps the motor up to a value ‘target Rpm”, then ramps it back down to idle (or to a specified rpm)
    - The ramping is done by incrementing a value: pulse. Then writing the motor to that value. The ramping speed and smoothness can be changed by adjusting the increment value (pulse += 5 and pulse -= 5) or changing the short delay (currently 50ms), making it slower or faster.
    - There is a delay in between ramping up and ramping back down, delay(MOTOR\_SPIN\_COATING\_TIME), this delay can be changed in settings depending on how long the spinning should occur.
      * Call Location: loop().

### Main\_SyringePump.cpp

* This contains the functions for controlling the syringe pump
* NOTE: This control can only turn on and off the pump. This is done by writing the device’s trigger to HIGH or LOW. Controlling the steps of the movement or stopping it must be done on the device’s keypad.
  + syringePump\_Setup()
    - The setup function for the syringe pump
      * Call Location: setup().
  + dispenseVolume()
    - This function turn on the pump. This is done by writing the trigger pin from logic HIGH (idle) to logic LOW (on), then after a short delay, back to logic HIGH. The device responds to this Falling Edge.



* + - NOTE: An adequate delay must be implemented to ensure the pumping sequence from the device is completed before moving on to further functions. The Arduino can not read information from the pump. Therefore, the pumping sequence must be done before, and take note of the time it takes.
      * Call Location: loop().

### Main\_LinearStage.cpp

* This contains the functions for controlling the linear stage.
* NOTE: There must be very precise calibration of the steps needed to move to positions. These calibrations can be done through iterative testing.
* Libraries Used: <AccelStepper.h>
  + linearStage\_Setup()
    - The setup function for the linear stage. This initializes all the pins, writes the ENABLE pin to HIGH (enabling the motor), and calls the homeStage() a, the function that homes the stage, and establishes a ‘0’ point for future reference.
    - Call Location: setup().
  + homeStage()
    - This is the homing function for the linear stage. The function sends the stage indefinitely toward a limit switch that is positioned at the end of the lead-screw. When the limit switch is hit the motor stops and sets the position as the ‘0’ or reference position.
    - Next Step: find the amount of steps that it takes to reach the far-right end of the stage. Set this position, x, as a soft-stop, and lower the speed when it gets to position x-100 steps.
      * Call Location: linearStage\_Setup().
  + moveToPosition()
    - The function that moves the linear stage to a predetermined position.
    - From the reference position (position ‘0’), the stage moves to either position A or Position B. These positions must be found to the exact step distance so that each capillary is directly over the centre of the coating platform.
      * Call Location: loop().