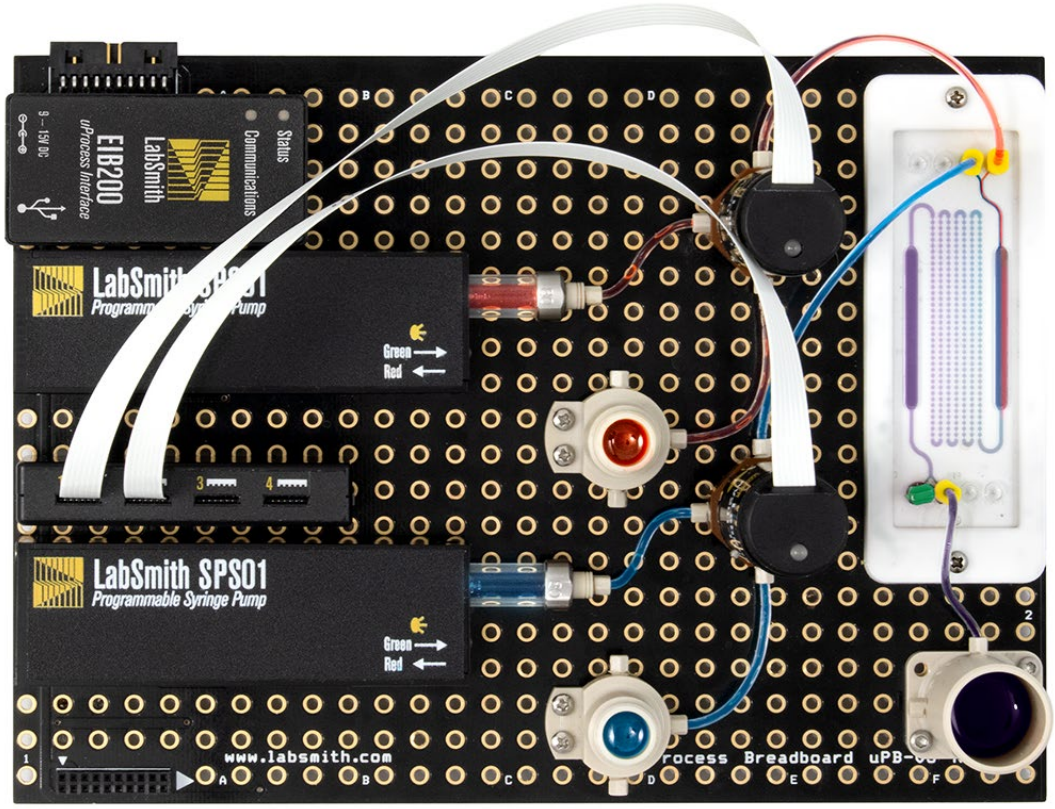
# Microfluidic Education Kit
# Assembly and Automation Instructions

**Co-flow of liquids across a microfluidic chip**




**Create a microfluidic controller to flow liquids across a microfluidic chip.**

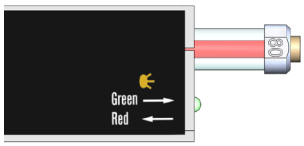
1. **Assemble fluidic circuits and attach to breadboard.**
2. **Connect to software**
3. **Create an automated program to control fluid flow**
4. **Observe flow on chip**

## Lab Equipment:

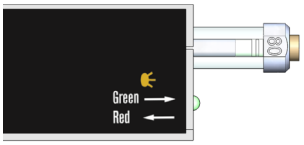
In this lab, you will assemble and automate a flow controller to demonstrate a fundamental property of microfluidics. Each of the individual components assembled on the breadboard has a unique function that helps get to the final goal: to deliver the dye from the 1 mL reservoirs through the chip.

### *Syringe pump*

A syringe pump is used to pull or push a fluid. The syringe pumps in this kit have a glass tube with a plunger inside. The plunger is moved backwards or forwards to fill or empty the syringe pump.




*Full Syringe pump*




*Empty syringe pump*

### *Reservoir*

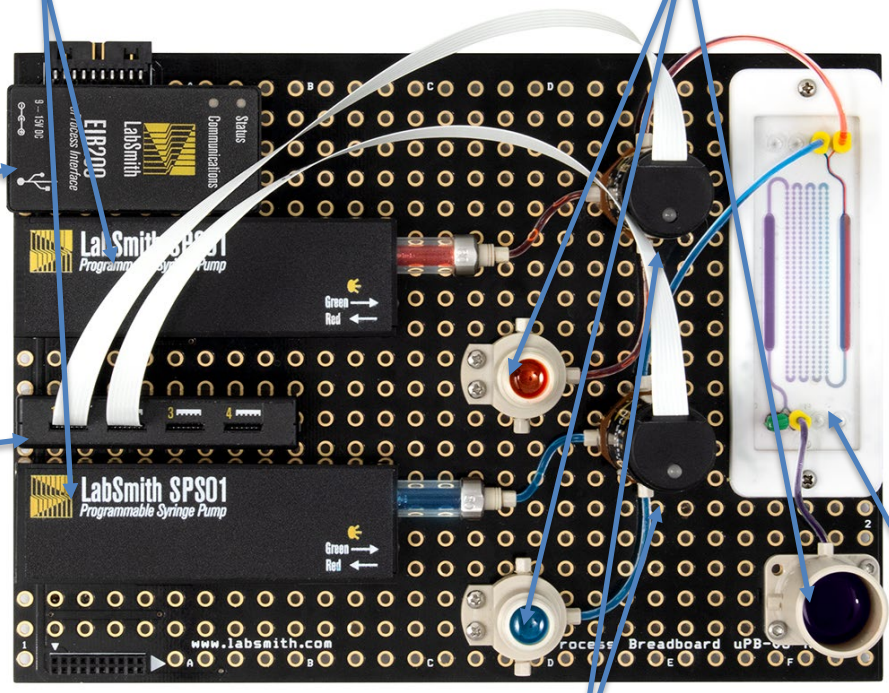
A reservoir is used to hold liquid. In this lab, the small reservoirs (1 mL) are filled with red and blue dye. The large reservoir (5 mL) will store the combined liquid after it goes through the microfluidic chip.

### *EIB200*

The EIB200 connects to power and to the computer so the instruments can be controlled with the software.

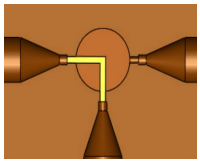
### *4VM02*

The 4VM02 connects to the valves so you can control them using the software.

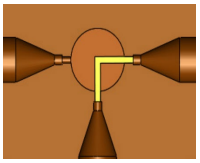



### *Valve*

A valve is used to direct the fluid flow. The valves in this kit have 3 ports and direct the flow from the center-to-left port, or from the center-to-right port.



*Valve in Position A*



*Valve in Position B*

### *Microfluidic Chip*

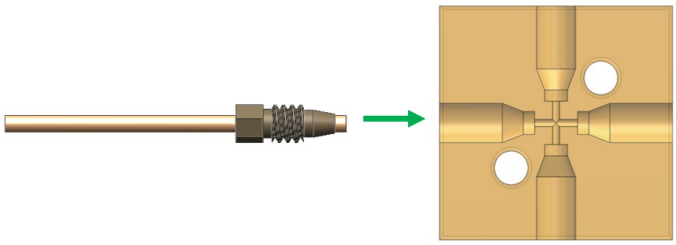
A microfluidic chip is a set of micro-channels etched or molded into a material. In this lab, the microfluidic chip has two input channels that come together in a Y formation.

**Pre-Lab Question:** In this exercise, the objective is to push the dye from the small reservoirs through the microfluidic chip and into the large reservoir. How would you accomplish this using the syringe pumps and valves?

## One-Piece Fitting Assembly Instructions

Before starting, it's important to understand how to properly connect the tubing to various ports using the one-piece fitting. When connecting tubing to a port using the one-piece fitting, start by inserting the tubing into the one-piece fitting. The tubing should protrude through the end of the fitting past the threaded section.

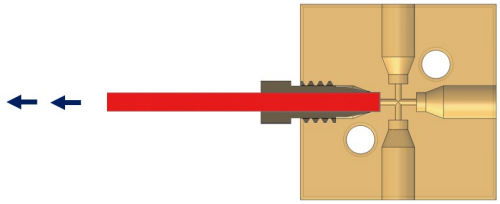
Screw the fitting and tubing into the port (see diagram below) until it is hand tight. Ensure that the tubing is securely seated into the port while tightening.




**Note: For a proper seal, push the tubing into the port while twisting the fitting in at the same time.**

Gently tug on the tubing to test that it is secure. If the tubing comes out, remove the fitting and repeat the steps above, verifying that the tubing extends past the end of the one-piece fitting (as shown in the diagram above). A loose connection will leak and must be fixed.

If the tubing still comes loose, use one of the 2 provided wrenches to tighten the fitting a quarter turn. Be careful not to break the fitting by overtightening it with the wrench.



See video at https://youtu.be/fV8R_ekezhE for more instructions.

| | **Tubing Lengths to Scale (for cutting reference)** |
|---|---|
| Short Tubing (4.5 cm) | |
| Medium Tubing (5.5 cm) | |
| Long Tubing (10 cm) | |

## Breadboard Assembly

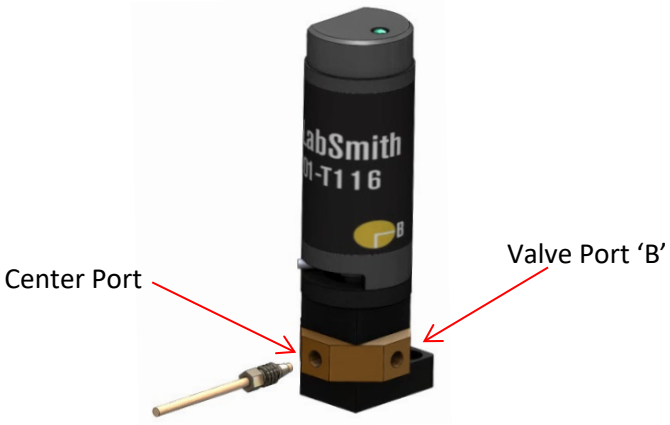
**Step 1: Install Valve Tubing**

| Valve | One-piece Fitting | Tubing (Short, Medium, and Long) |
|---|---|---|
|  |  |  |
| 2x | 6x | 2x (4.5 cm), 2x (5.5 cm), 2x (10 cm) |

**Note**: Two identical assemblies will be made in this step. If working in groups, these can be assembled at the same time.

1.1 Cut two pieces of short tubing (4.5 cm) with sharp scissors or a blade (refer to page 3 for a to-scale cutting reference with each tube length). With a one-piece fitting, connect a piece of short tubing to the center port of the valve (see diagram below). Repeat for both valves.

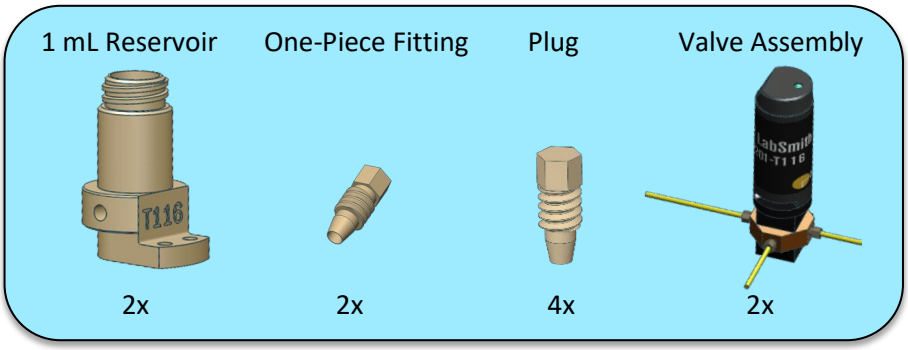
**Note**: See previous page for tubing lengths to scale if you don't have a ruler.




1.2 Cut two pieces of medium tubing (5.5 cm). With a one-piece fitting, connect a piece of medium tubing (5.5 cm) to valve port 'B' (see valve label or diagram above). Repeat for both valves.

1.3 Cut two pieces of long tubing (10 cm) and secure a piece of long tubing to the third port (port A) on the valve with a one-piece fitting. Repeat for both valves. When finished, you should have 2 identical valve assemblies.

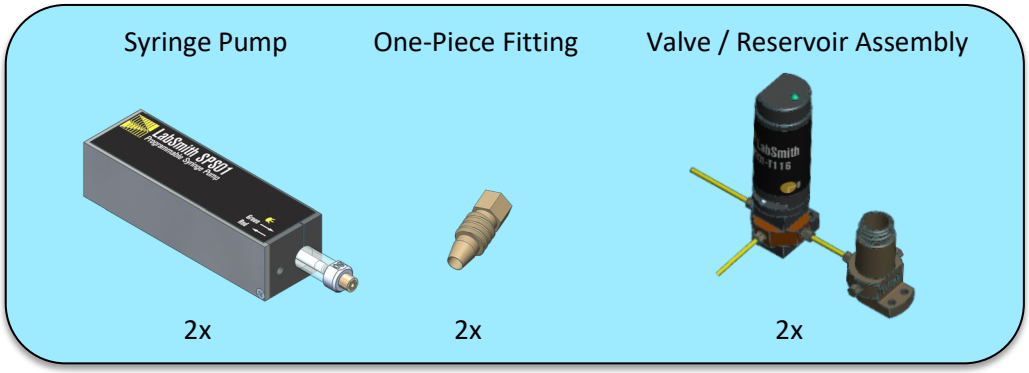
**Step 2: Install Intake Reservoirs**



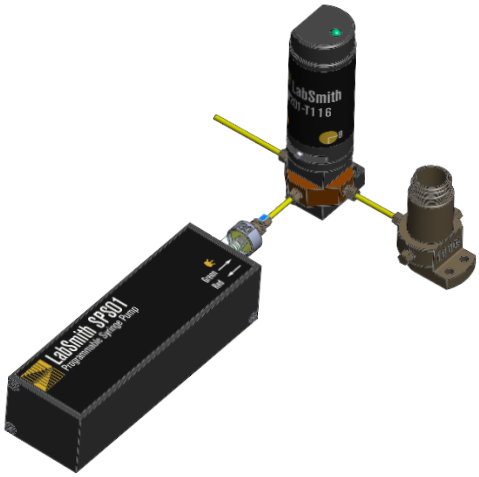

2.1 Insert a plug into the each of the side ports of both 1 mL reservoirs, leaving the center port open.

2.2 Connect the tubing from valve port 'B' to the open reservoir port using a one-piece fitting. Repeat for both valves.

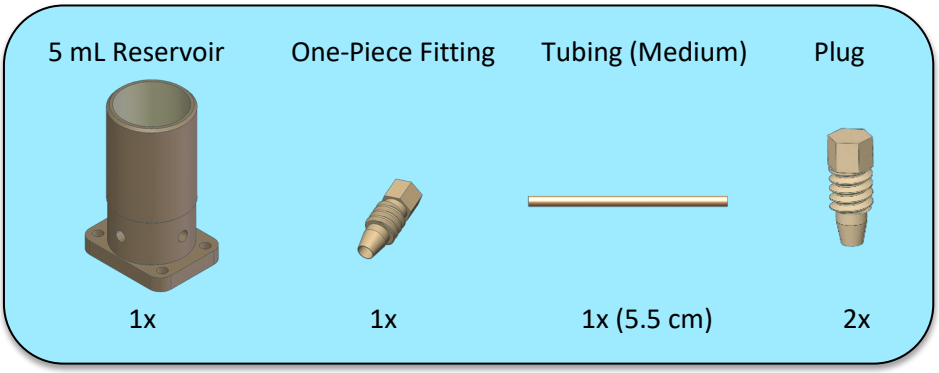

**Step 3: Install Syringes**




3.1 Connect the tubing from the center valve port to the syringe pump with a one-piece fitting. Complete this for both valve assemblies.

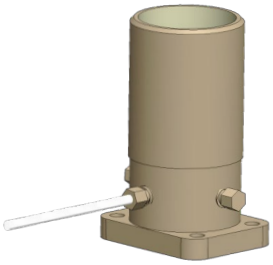


**Step 4: Assemble Waste Reservoir**

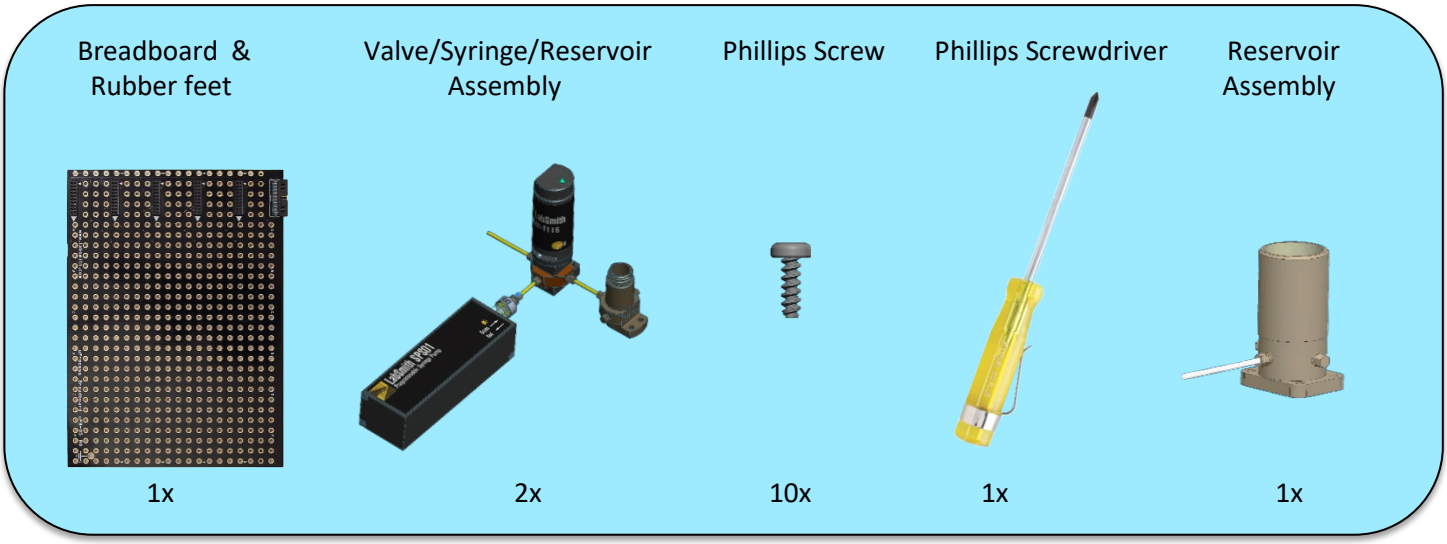



4.1 Insert a plug into the each of the side ports of the 5 mL reservoir, leaving the center port open.

4.2 Cut a piece of medium tubing (5.5 cm). Connect the piece of medium tubing (5.5 cm) to the open center port on the 5 mL reservoir using a one-piece fitting.



**Step 5: Secure the assemblies to a Breadboard**

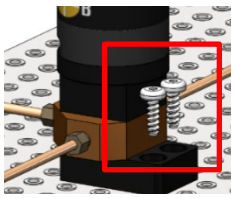



**Note**: See the picture below for approximate breadboard layout suggestions.

5.1 Attach a rubber foot (included with the breadboard) at each of the 4 corners on the bottom of the breadboard.

5.2 Plug a syringe pump (connected to the valve and reservoir) into a connector on the side of the breadboard. (Notice there is a slot on the bottom of the syringe that the connector fits into).

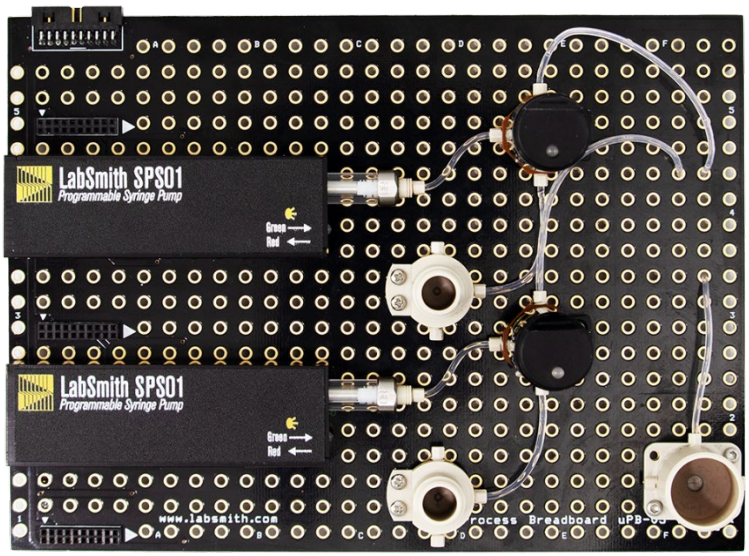
5.3 Connect the valve to the breadboard using 2 screws (see diagram below).



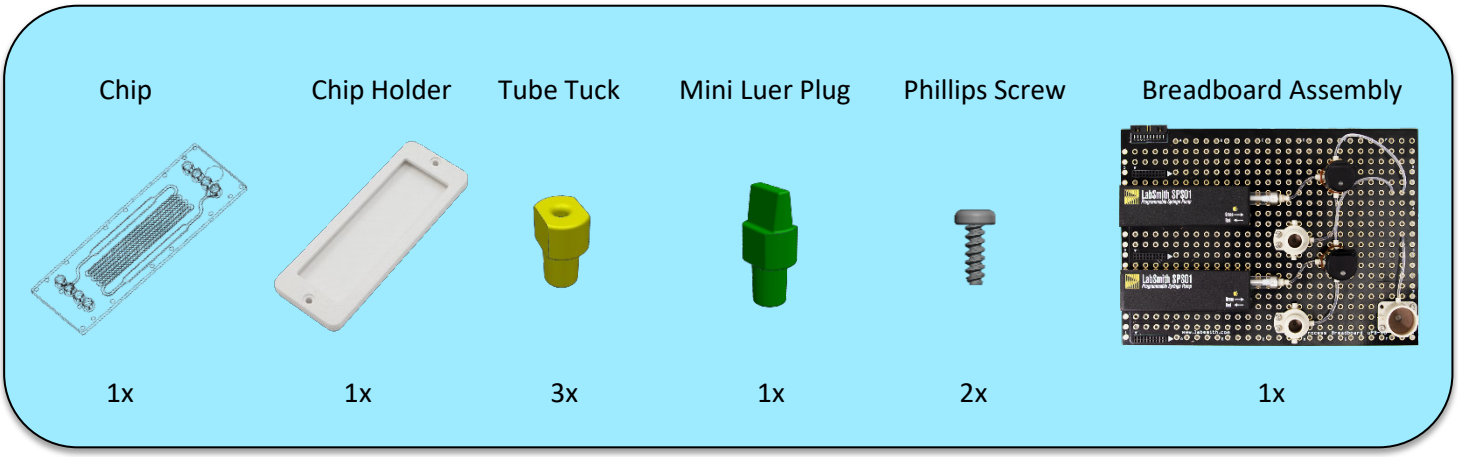
5.4 Connect the 1 mL reservoir to the breadboard using 2 screws.

5.5 Repeat steps 1–3 for the second valve-syringe-reservoir assembly.

5.6 Connect the 5 mL reservoir assembly to the board using 2 screws. The tubing connected to the center port of the 5 mL reservoir should face inwards on the breadboard.




**Step 6: Connect the Chip**

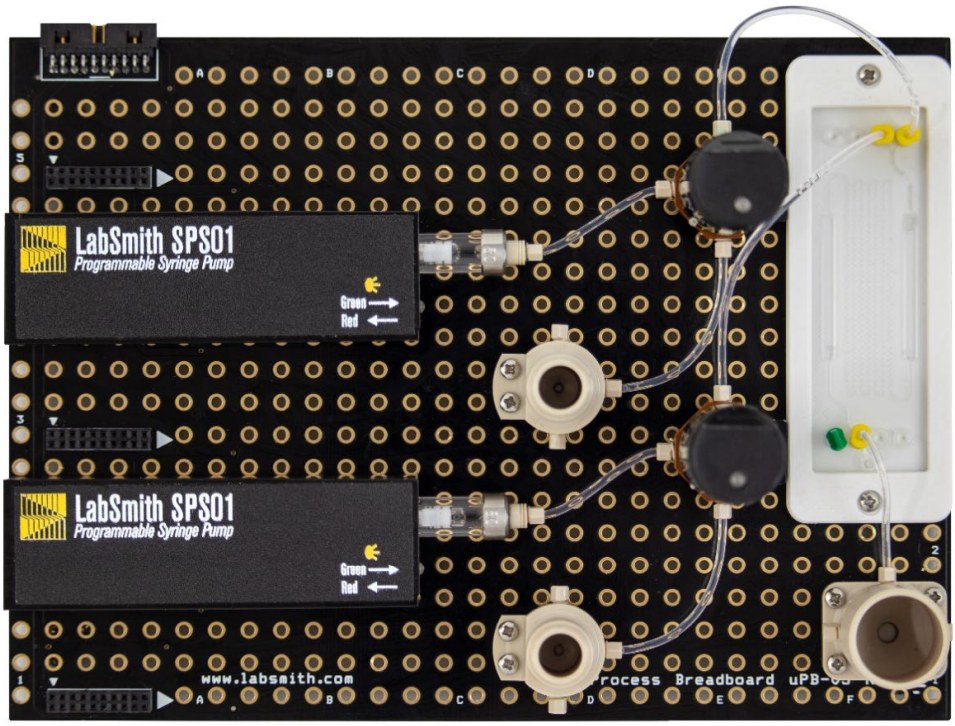



6.1 Snap the bottom and top pieces of the chip holder over the chip.

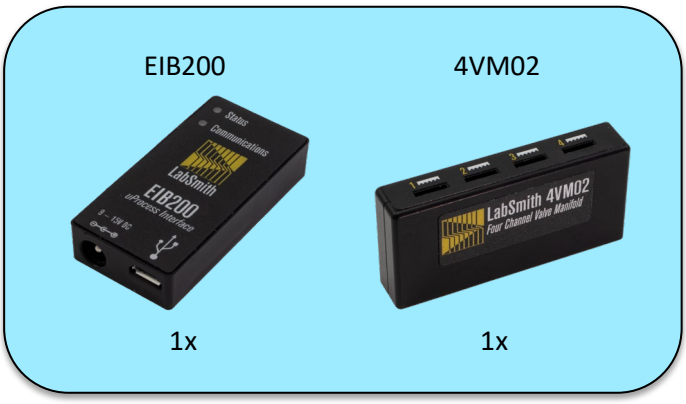
6.2 Insert the tubing from the 5 mL reservoir into the large end of a tube tuck connector. Then, place the narrow end of the tube tuck connector (with the tubing from the 5 mL reservoir now inserted) into one of the outlet ports on the chip near the waste reservoir. Plug the other outlet port with the included mini Luer plug. Only two out of four ports will be used on each side of the chip. The other two ports on each side will not be used.

6.3 Insert the tubing from each of the 2 valves into 2 tube tuck connectors. Then, place the narrow end of the tube tuck connectors into the chip's remaining 2 intake ports. Screw the chip to the breadboard (see the picture below for the recommended location).

**CAUTION**: The microfluidic chip's intake and outlet ports are fragile. Always insert tubing into the tube tuck connector before inserting into the chip.

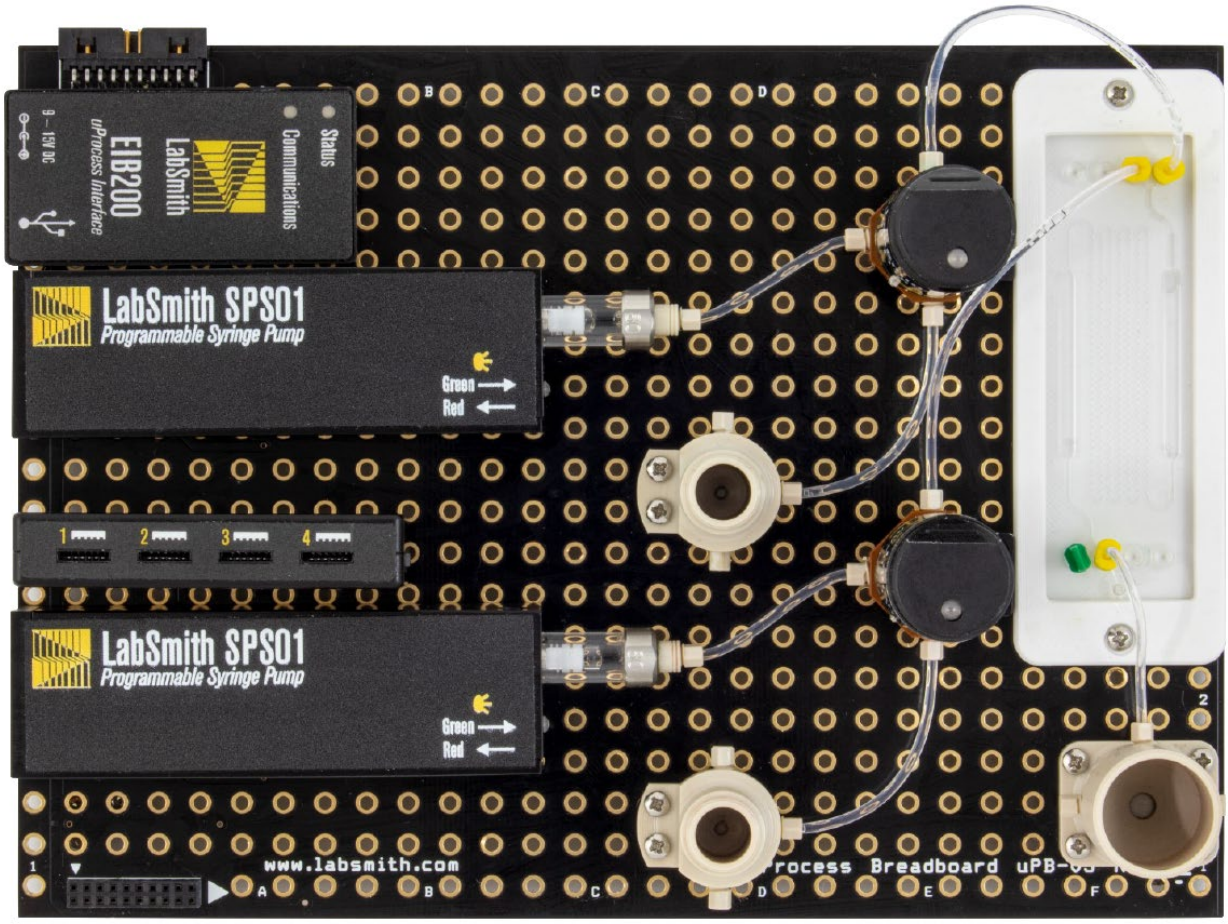



**Step 7: Install the EIB200 and 4VM02**

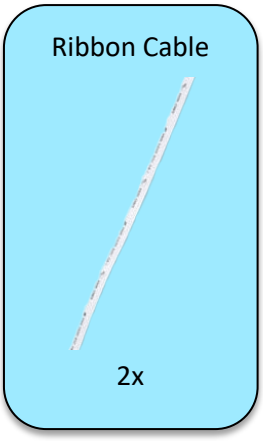


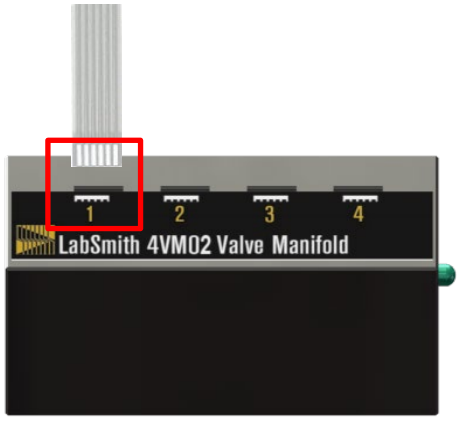

7.1 Connect the EIB200 and 4VM02 to connectors on the left side of the breadboard. See the picture below for recommended locations.

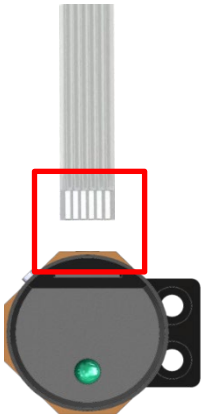
Completed breadboard:




**Step 8: Connect Cables**



8.1 Use the two ribbon cables to connect the 4VM02 (valve manifold) to the two valves. Refer to the labels on the 4VM02 and the valves for the correct orientation of the cable. Both ends must be inserted correctly for the valve to function.

## Connect to Software

**Step 9: Software and Driver Installation:**

NOTE: *If your computer is already equipped with uProcess and the necessary USB drivers, skip to Step 10.*

9.1 **Install the uProcess software**. The software can be installed from the USB flash drive included in the kit (go to *Software Installation>setup.exe*), or downloaded from our website https://labsmith.com/downloads/download-uprocess-software/

9.2 **Install the USB driver**. From the flash drive go to *Software Installation>CP210x Windows Drivers* and run one of the following files:

a. **CP210xVCPInstaller_x64.exe** if you have a Windows x64 computer
b. **CP210xVCPInstaller_x86.exe** if you have a Windows x32 computer

**NOTE**: This step may not be required for newer operating systems.

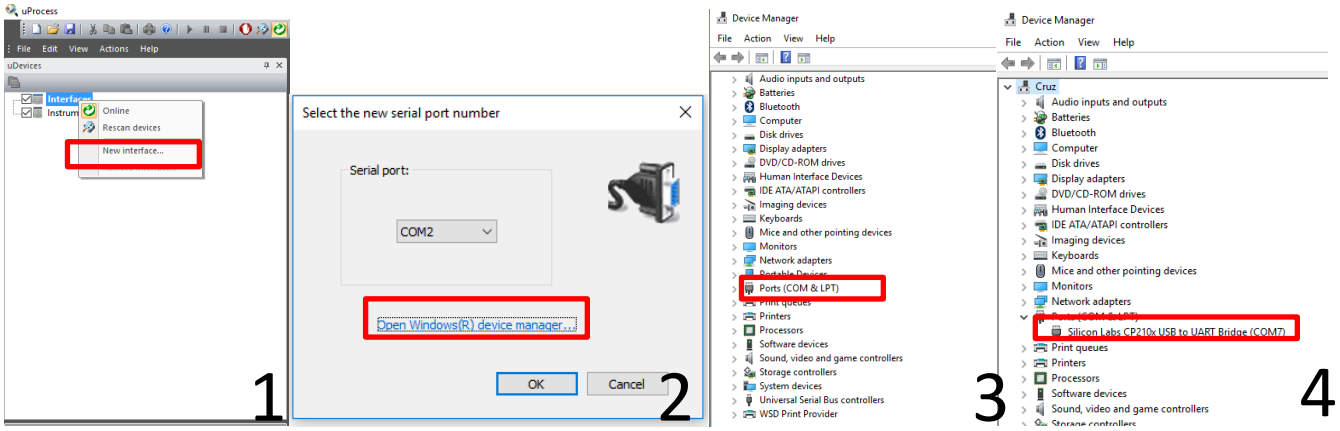
**Step 10: Connect the Breadboard to the Computer**

10.1 Connect the power cable to an outlet and the micro-USB connector into the computer. Plug the power and the micro-USB into the EIB200.

10.2 Open the uProcess software.

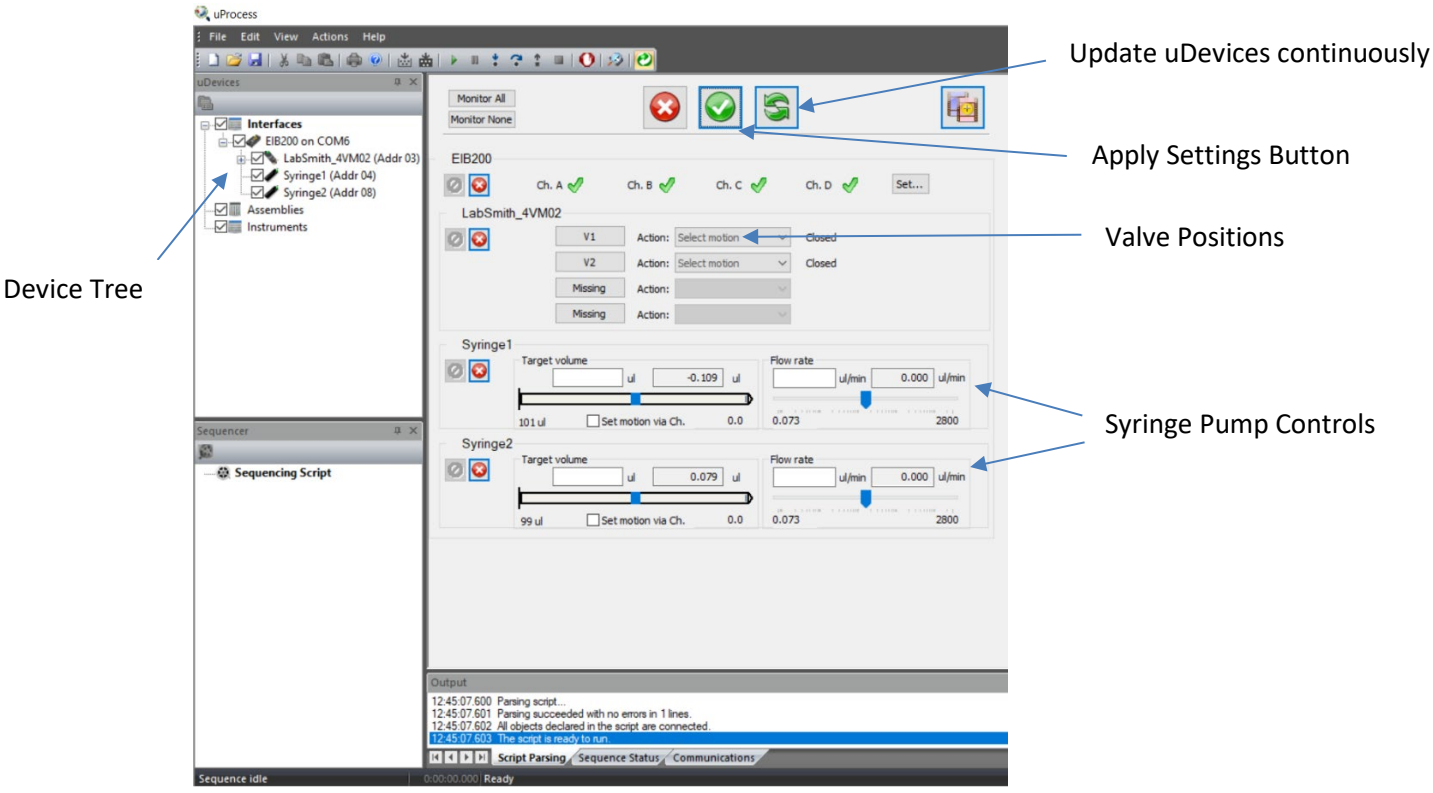

**uProcess Icon**

10.3 Right-click on '*Interfaces*' and select '*New Interface.*' Then select the corresponding COM serial port from the drop-down menu. If the COM port is unknown, click on '*Open Windows device manager…*' and view the Ports list. The correct COM port is listed as '*Silicon Labs CP210x USB to UART Bridge.*' The COM port will be listed at the end of the line.



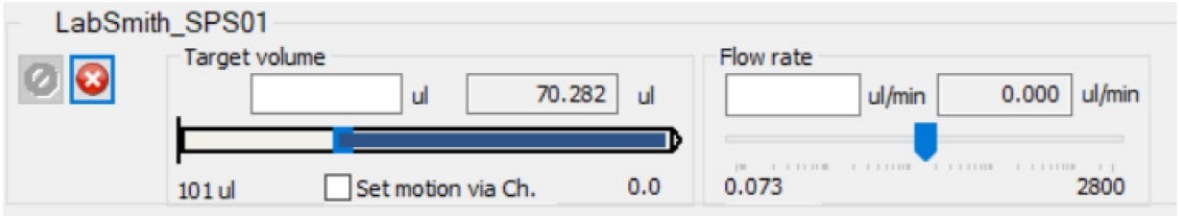
**Step 11: Manually Controlling the Syringe Pumps and Valves**

11.1 Select 'Interface' in the upper left-hand corner to open up this window:



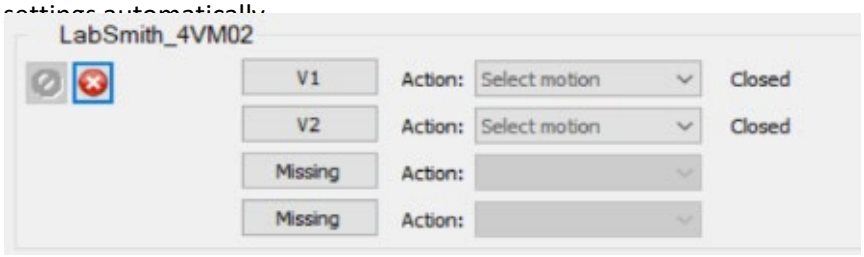

**Syringe Pumps:** The syringe pumps are used to push or pull fluids through your microfluidic system. Each syringe pump has a glass tube with a plunger inside. The software is used to move the plunger to push and pull the fluids.

11.2 Experiment with the syringe pump automation by adjusting the slider bars on one of the syringe pump controllers to change the syringe plunger location and speed. Press the '*Apply Settings Now*' button ( ) to execute changes.




**Valves:** The valves are used to direct the flow in your microfluidic system. The valves have 3 ports: A, B and Center. In Valve position 'A', ports A and Center are connected. In valve position 'B', ports B and Center are connected.

11.3 Experiment with the valve controls by changing the valve positions and pressing the '*Apply Settings Now*' button ( ) to execute changes. Alternatively, select '*Update uDevices Continuously*' ( ) to apply the settings automatically.




**Note:** If your valves are listed as missing on the 4VM, ensure that the ribbon cables are connected with the correct orientation and click *'Rescan for Devices'* ( ).

11.4 Leak Check: Once you have demonstrated that you can actuate the syringe pumps and valves, fill the two 1 mL intake reservoirs ¾ full with distilled or filtered water.

11.5 Try filling one of the syringes with water from the reservoir. You'll need to set the valve so it is open to the reservoir and then actuate the syringe pump's plunger to fill the syringe.

11.6 Once you have filled the syringe, push the water through the chip. You will have to open the valve towards the chip and move the water through. Verify that both flow streams work with no leaks. If there is a problem, check the troubleshooting guide.

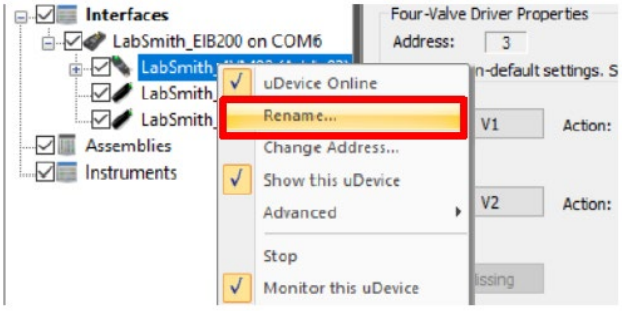
## Automate your Microfluidic Controller

In the next steps you will create an automated script to pump fluids across your microfluidic chip. You will first set the valve position so you can pull liquid from the 1 mL reservoirs to fill the syringes, then change the valve position to push the liquid across the microfluidic chip.

### Step 12: Renaming uDevices and Valves

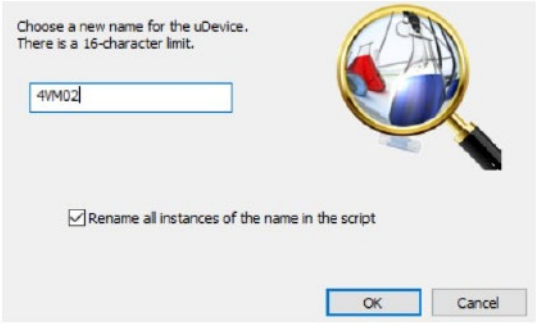
12.1 Before automating the microfluidic system with a script, each device needs a unique name. The two syringes named 'LabSmith_SPS01' will be renamed to 'Red' and 'Blue' (to represent the red and blue dye that will move through the system), and the device named 'LabSmith_4VM02' can be renamed to '4VM02' for simplicity.

a. To rename the devices accordingly, right click on the device to be renamed and select '*Rename*…'



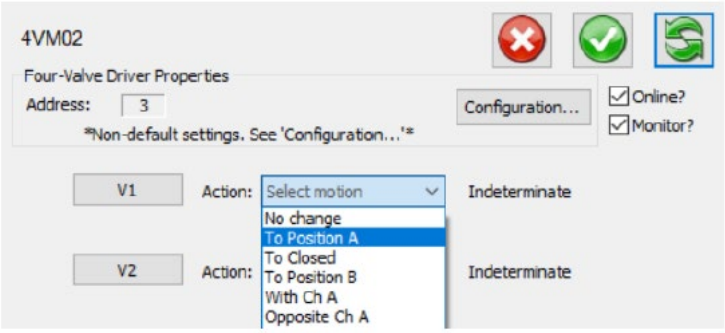

b. Rename one syringe pump to 'Red' and the other to 'Blue'. To determine which syringe pump was named 'Red' or 'Blue', click on one of the syringes under the device tree. An LED light will flash on the side of this syringe pump. Observe which syringe pumps were renamed to 'Red' and 'Blue' because the valves connected to each syringe pump will be renamed alike.

c. Rename the device called ‘LabSmith_4VM02’ to ‘4VM02’.



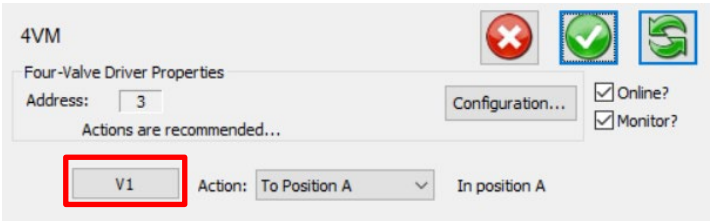

12.2 To determine which valve is connected to the ‘Red’ or ‘Blue’ syringe pumps, change the position of one of the valves and observe which valve is actuated.

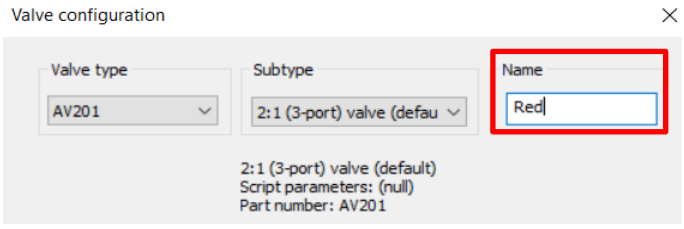
a. With the 4VM02 selected under the device tree, click the drop-down menu next to one of the valves and select ‘*To Position A*’ or *‘To Position B*’. The particular position does not matter during this step.




b. Click the ‘*Apply Settings Now’* button ( ) and the selected valve will actuate. Determine if this valve is connected with the syringe pump named ‘Red’ or the syringe pump named ‘Blue’.

c. To rename the valves, click on the name of the valve to the left of the position drop-down menu. This will open the valve configuration box, where the valve can be renamed under ‘Name’. Rename the valves either ‘Red’ or ‘Blue’ according to the name of the syringe pump each valve is connected to.

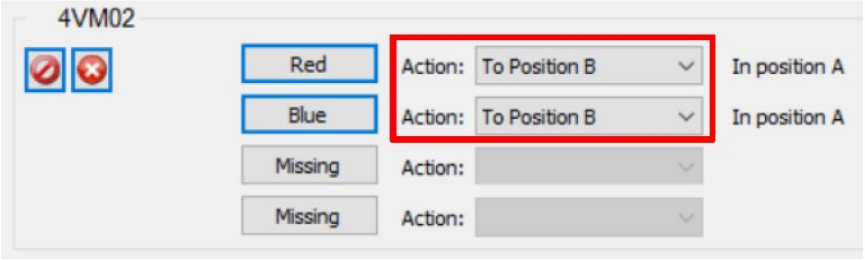



**Step 13: Set Valves**

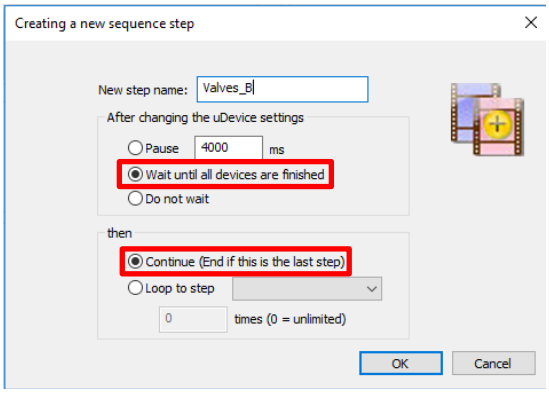
13.1 Make sure ‘*Update uDevices Continuously*’ ( ) is NOT selected by clicking on the ‘*Stop All uDevices’ Button* ( ). Then, click on ‘Interfaces’ in the upper left of your window.

13.2 First, automate the valves to open to position ‘B.’ Select ‘To Position B’ in the 4VM02 box for both valves.

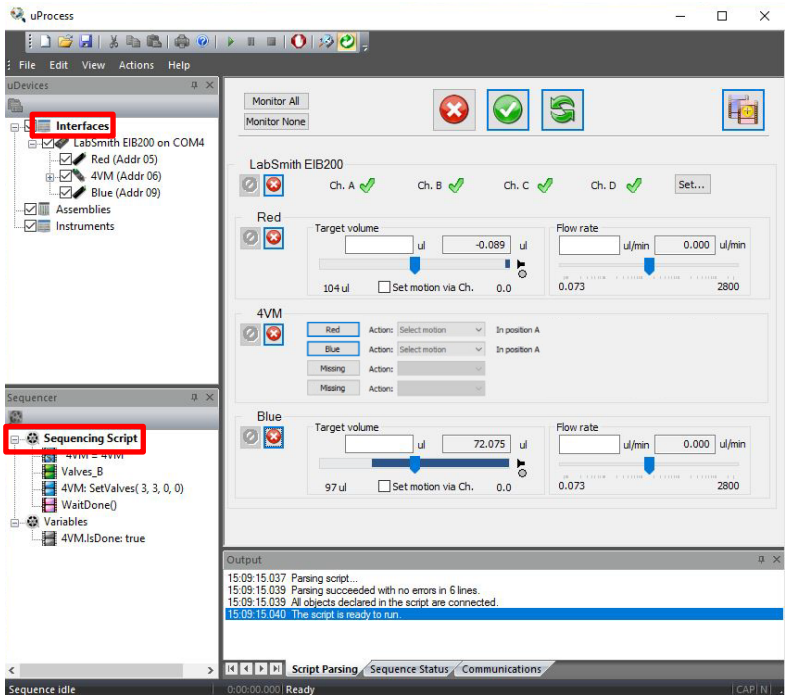



Do NOT select the ‘*Apply Settings Now*’ button ( ) or you will have to repeat these steps.

13.3 Click the '*Save as New Sequence Step*' button (  ), which will only be visible when *'Interfaces'* is selected in the upper left of your window. Name the step '*Valves_B*' and ensure '*Wait until all devices are finished*' and '*Continue*' are selected.




13.4 Click on 'Sequencing Script' in the lower lefthand corner to open the script editing window.




13.5 Your script should have the following step that will open the valves to the 'B' position.

```
*4VM     =        4VM

Valves_B:
        4VM:     SetValves( 3, 3, 0, 0)
        WaitDone()
```

Declaration heading for the 4VM

These numbers represent the position of the valve.
0 = no change
1 = open valve to 'A'
2 = close valve
3 = open valve to 'B'
Since you only have two valves plugged in, the other two will be set to zero. The order of valves is the order they are plugged into the 4VM.

**Step 14: Fill Syringes**

14.1 Click on 'Interfaces' in the upper left of your window.

14.2 Set both syringe pumps' target volume to 80 ul with a fill rate of 300 ul/min. Enter these values under '*Target Volume*' and '*Flow Rate*,' respectively.


14.3 Click the '*Save as New Sequence Step*' button (  ). Name the step '*Fill_Syringes*' and ensure '*Wait until all devices are finished*' and '*Continue*' are selected.

14.4 Again, click on the 'Sequencing Script' in the lower left corner. It will now include code instructing the syringes to fill with water.

```
Fill_Syringes:
        Red:    SetFlowRate( 300.000 ul/min)
        Red:    MoveTo( 80.000 ul)
        Blue:   SetFlowRate( 300.000 ul/min)
        Blue:   MoveTo( 80.000 ul)
        WaitDone()
```

14.5 Now that you have created code to fill the syringe, you will program the valves and syringes to push the water through the chip.

**Step 15: Switch Valves**

15.1 Click on 'Interfaces'. Set both valves to '*Open A*' and click the '*Save as New Sequence Step*' button (  ). Name the step '*Valves_A*' and ensure '*Wait until all devices are finished*' and '*Continue*' are selected.

15.2 Your script will have added the following step:

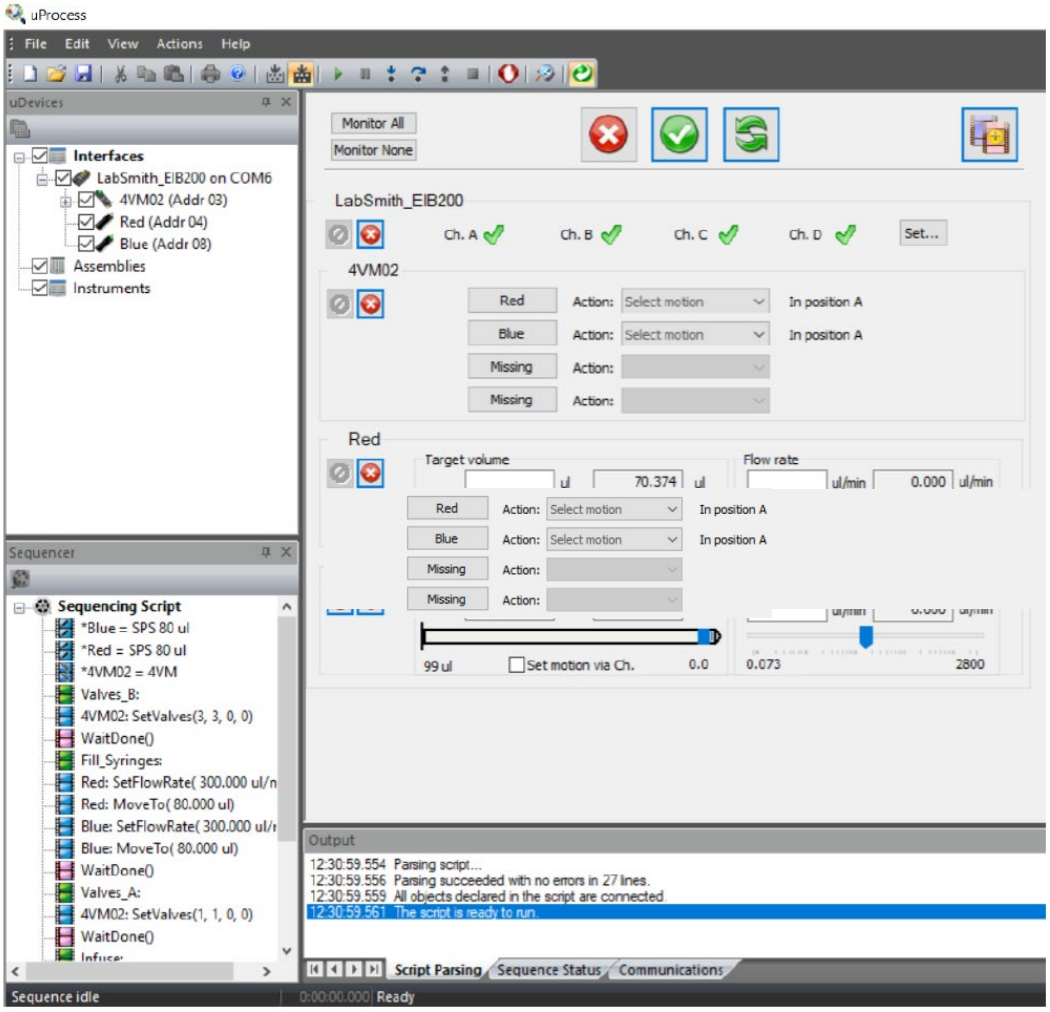
```
Valves_A:
        4VM02: SetValves(1, 1, 0, 0)
        WaitDone()
```

**Step 16: Infuse**

16.1 Click on *'Interfaces'*. Set both syringe pumps' target volume to 0.5 ul with a flow rate of 100 ul/min.

16.2 Click the '*Save as New Sequence Step*' button (  ). Name the step '*Infuse*' and ensure '*Wait until all devices are finished*' is selected. Then select '*Loop to Step*', select '*Valves_B'* from the dropdown menu, and set the number of *'times'* to 5.

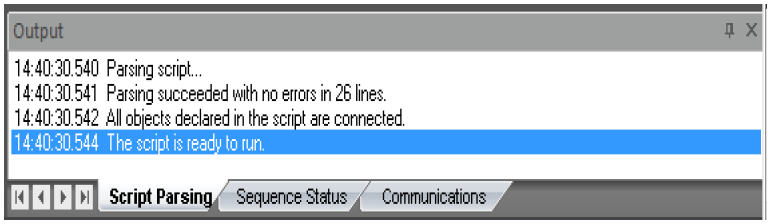
16.3 Your window should now look like the following:




**Step 17: Check Script**

17.1 Click on '*Sequencing Script*'. This will display the automated script that has been created. You can edit the script in this window.

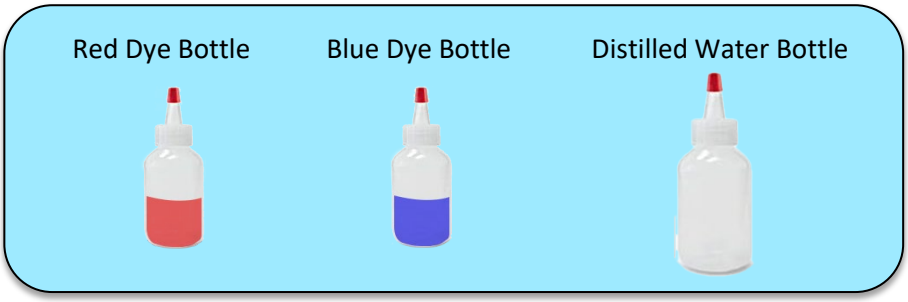
**Note:** If there are errors in the script it won't run. All errors are described in the output window beneath the script. Double clicking on the error will highlight the line of the error.




This script will move the valves to position B, fill the syringe pumps, move the valves to position A, and push the liquid through the chip. It will complete this cycle 5 times. Adding dye to the system will help to observe the liquids mixing in the chip.

```
*Blue   =       SPS     80 ul
*Red    =       SPS     80 ul
*4VM02  =       4VM

Valves_B:
        4VM02:  SetValves(3, 3, 0, 0)
        WaitDone()

Fill_Syringes:
        Red:    SetFlowRate( 300.000 ul/min)
        Red:    MoveTo( 80.000 ul)
        Blue:   SetFlowRate( 300.000 ul/min)
        Blue:   MoveTo( 80.000 ul)
        WaitDone()

Valves_A:
        4VM02:  SetValves(1, 1, 0, 0)
        WaitDone()

Infuse:
        Red:    SetFlowRate( 100.000 ul/min)
        Red:    MoveTo( 0.500 ul)
        Blue:   SetFlowRate( 100.000 ul/min)
        Blue:   MoveTo( 0.500 ul)
        WaitDone()
        Loop Valves_B  5
```

**Step 18: Fill the Reservoirs**

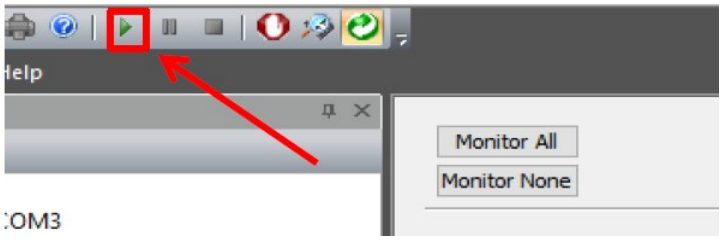



18.1 Fill all three bottles with distilled (or filtered) water and add 5-10 drops of food coloring to each of the small (1 oz) bottles. Red and blue coloring is recommended.

18.2 Add the distilled water to both intake reservoirs (after checking for leaks you will add the dyed water).

**Step 19: Run Automation**

19.1 Click the green '*Run*' button in the uProcess toolbar.




To stop the automation at any point, click the '*Stop*' button ( ) in the uProcess toolbar. The process can be restarted by pressing the '*Run*' button ( ).

Save your Script by clicking the save button ( ) or using the shortcut Ctrl + S.

19.2 The process will run for 5 cycles.

19.3 Fix any leaks or problems.

19.4 Once the operation has run a few times and there are no leaks, empty the reservoirs and fill with red- and blue-dyed water (according to how the valves/syringe pumps were renamed).

19.5 Run the operation again and write down your observations. You may have to let the automation run several times before you can make observations, especially if you have bubbles in your tubing.

While the script is running, the '*Sequencing Script*' will highlight the current line. Clicking on '*Interfaces*' will show you the status and position of each device.

19.6 Try changing the flow rate and volume for the red and blue dye liquids (click on the script tree in the lower left corner of the uProcess window to make changes to the script). Run the operation and note any changes or observations. Refill the intake reservoirs as needed.

```
Valves_B:
      4VM02: SetValves(3, 3, 0, 0)
      WaitDone()
Fill_Syringes:
      Red:   SetFlowRate( 300.000 ul/min)
      Red:   MoveTo( 80.000 ul)
      Blue:  SetFlowRate( 300.000 ul/min)
      Blue:  MoveTo( 80.000 ul)
      WaitDone()
Valves_A:
      4VM02: SetValves(1, 1, 0, 0)
      WaitDone()
Infuse:
      Red:   SetFlowRate( 100.000 ul/min)
      Red:   MoveTo( 0.500 ul)
      Blue:  SetFlowRate( 100.000 ul/min)
      Blue:  MoveTo( 0.500 ul)
      WaitDone()
      Loop Valves_B5
```

Syringe fill volume

Dye flow rate

# Flow Manipulation Exercises

**Useful Script Commands**

*Break* *Break*

Automatically activates the pause toolbar button, pausing Sequence until the pause toolbar button is pressed. This function is useful for testing a script, or if you have an event with unpredictable timing that needs to be controlled manually.

Example: `Break`

Note: The Break function cannot be added using the uProcess Wizard

*Goto* *Goto* **StepName**

Creates a loop back to named step.

*StepName* *defined name of step in script*

Example: `Goto Infuse`

*Loop* *Loop* **StepName cycles**

Loops back to named step for specified number of cycles.

*StepName:* defined name of step in script

*cycles:* number of times to loop

Example: `Loop Infuse 6`

Note: For an endless loop, use the **Goto** command.

*Log* *log*(**on**)
*log*(**off**)
*log*(**filename**)

Opens/closes a log file.

*filename:* file path and name (see examples)

If omitted, script will prompt for file name. File naming can be specified in *View>Status/Meas Logging Options…*

Note: If the file name or path contains a space, the entire name and path must be surrounded by double quotes (see example)

Note: A Script Error will occur and the script will halt if a **log (on)** command is issued when a log file is already open or a **log (off)** command is issued when a log file is not open

Note: If you include a directory, make sure it exists and that you have access

Examples:
```
log (on)
log (C:\Temp\testlog.csv)
log ("C:\Temp\test log.csv")
log (testlog)
log (off)
```

*MoveTo* **SyringeName***: MoveTo (***vol units**)

Sets the desired syringe position.

*SyringeName* defined name of syringe (“Red” or “Blue”)

*vol* desired syringe volume

*units* volume units (ul)

Example: `Red: MoveTo( 80 ul)`

*Wait* *Wait* (**duration units**)

Pauses script for specified duration.

*Duration* length of pause

*units* pause duration units (seconds, s)

Example: `Wait( 1 s)`

*WaitDone* *WaitDone()*

Pauses script until current command is completed.

Example: `WaitDone()`

*SetFlowRate* **SyringeName***: SetFlowRate (***rate units**)

Sets syringe flow rate.

| | | |
|---|---|---|
| *SyringeName* | | defined name of syringe (must be defined in script header) |
| *rate* | | syringe flow rate (positive number) |
| *units* | | flow rate units (ul/min) |
| Example: | `Red:` | `SetFlowRate( 50 ul/min)` |

*SetValves* **4VM02Name***: SetValves(* **x, x, x, x**)

Sets valve positions.

| | | |
|---|---|---|
| *4VM02Name* | | defined name of 4VM02 (in this case "4VM02") |
| *x, x, x, x* | | position of valves 1, 2, 3, and, 4, respectively, where<br>x = 0 no change<br>x = 1: open valve to 'A'<br>x = 2: close valve<br>x = 3 open valve to 'B' |
| Example: | `4VM02:` | `SetValves( 1, 0, 3, 2)` |

*Stop* **DeviceName***: Stop()*

Stops device.

| | | |
|---|---|---|
| *DeviceName* | | defined name of 4VM02 or syringe (must be defined in script header) |
| Example: | `Red:` | `Stop()` |

*If Statements*

An if statement will run code if a condition is true. There are three ways to format an if statement.

If (*condition*)
    *Statement* ; this only works if there is only one statement to run
If (*condition*)
{
    *Statement* ; with brackets, you can have more than one statement that runs condition is true
}
x = (*condition*)? *value1* : *value2* ; if condition is true, x is assigned *value1* if false, x is assigned *value2*
Examples:

```
If (x > y)
    x = y

If (x > y)
{
    x = y
}

x = (x>y)? y : z
```

*While Loops*

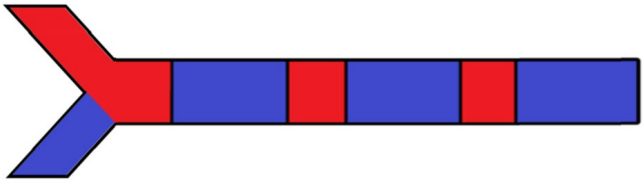
A while loop will repeat lines of code while a condition is true.

While (*condition*)
{
*Statement*
*Statement*
}
Example:

```
a = 1                ; this code will beep 10 times and finish
While (a <=10)
{
    Beep()
    Wait(1 s)
    a = a + 1
}
```

## Patterned Flow

In the following exercise, you will vary the red and blue flow to create a pattern on the chip.



Comprehension Questions:

1. How do you make the script wait for a set amount of time before moving on to the next line?
2. How do you stop a syringe pump from moving?
3. Automation is typically broken into steps. How could you break your code into repeatable steps?

Create the flow pattern shown above using the glossary of script functions.

## Gradient Flow

Next you will use variables in your code to create a gradient flow. You will write a script so that the chip will start out with all red dye and then slowly transition to all blue dye. Both Syringes should run out of dye at the same time.

Variables can be defined by math:

```
a = 7
b = a + 3
c = b - 5
d = c * 2
e = d / 5
```

Or reassigned to the same variable:

```
a = 7
a = a + 3
a = a - 5
a = a * 2
a = a / 5
```

You will use these variables to set flow rates and positions of the syringes. For example, the script below functions the same as the original script you wrote, except it uses variables to assign the flow rate.

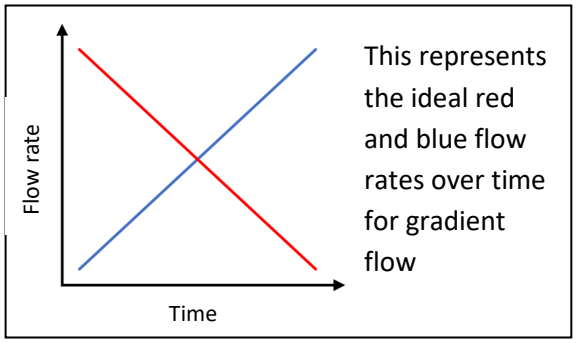
```
*Blue   =     SPS    80 ul
*Red    =     SPS    80 ul
*4VM02 =      4VM

      a = 100
      R1 = 3 * a
      R2 = a

Valves_B:
      4VM02: SetValves(3, 3, 0, 0)
      WaitDone()

Fill_Syringes:
      Red:   SetFlowRate( R1 ul/min)
      Red:   MoveTo( 80.000 ul)
      Blue:  SetFlowRate( R1 ul/min)
      Blue:  MoveTo( 80.000 ul)
      WaitDone()

Valves_A:
      4VM02: SetValves(1, 1, 0, 0)
      WaitDone()

Infuse:
      Red:   SetFlowRate( R2 ul/min)
      Red:   MoveTo( 0.500 ul)
      Blue:  SetFlowRate( R2 ul/min)
      Blue:  MoveTo( 0.500 ul)
      WaitDone()
      Loop Valves_B5
```

NOTE: Use the Script Wizard to create the basic code, then type in modifications as needed.

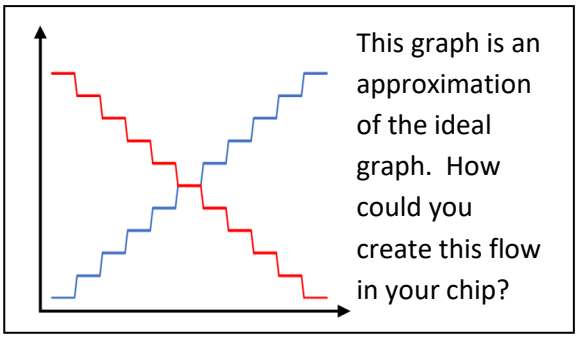
A ‘while’ loop will run the code between the brackets “{ }” while the condition (in this case “a < 10”) is true.

```
a = 0
while (a < 10)
{
a = a + 1
Wait(2 s)
Beep
}
```

You will now create a gradient flow. The objective is to begin by flowing only red dye and then slowly transition to only blue dye.




This represents the ideal red and blue flow rates over time for gradient flow



This graph is an approximation of the ideal graph. How could you create this flow in your chip?

This code would work, but it would take a long time to type in each step.

```
Infuse1
        Red:    SetFlowRate (100 ul/min)
        Red:    MoveTo (0.500 ul)
        Blue:   SetFlowRate (1 ul/min)
        Blue:   MoveTo (0.500 ul)
        Wait( 1 s)

Infuse2
        Red:    SetFlowRate (99 ul/min)
        Red:    MoveTo (0.500 ul)
        Blue:   SetFlowRate (2 ul/min)
        Blue:   MoveTo (0.500 ul)
        Wait( 1 s)
Infuse3
        Red:    SetFlowRate (98 ul/min)
        Red:    MoveTo (0.500 ul)
        Blue:   SetFlowRate (3 ul/min)
        Blue:   MoveTo (0.500 ul)
        Wait( 1 s)

        .
        .
        .
Infuse100
        Red:    SetFlowRate (1 ul/min)
        Red:    MoveTo (0.500 ul)
        Blue:   SetFlowRate (100 ul/min)
        Blue:   MoveTo (0.500 ul)
        Wait( 1 s)
```

How could you do this using math and a loop instead?

The following is an example of code with some values missing. Fill in the missing code so the chip will start out with all red dye and then slowly transition to all blue dye. Both Syringes should stop moving at the same time.

```
*Blue                   =       SPS01 80 ul
*Red                    =       SPS01 80 ul
*4VM                    =       4VM01

Valves_B:
        4VM:    SetValves (3, 3, 0, 0)

Fill_Syringes:
        Red:    SetFlowRate (300 ul/min)
        Red:    MoveTo (80.000 ul)
        Blue:   SetFlowRate (300 ul/min)
        Blue:   MoveTo (80.000 ul)
        WaitDone()

Valves_A:
        4VM:    SetValves (1, 1, 0, 0)
        WaitDone()

SetInitialValues:
        redFlow = ____
        blueFlow = ____

Infuse:
        while (________________)
        {
        Red:    SetFlowRate (redFlow ul/min)
        Red:    MoveTo (0.500 ul)
        Blue:   SetFlowRate (blueFlow ul/min)
        Blue:   MoveTo (0.500 ul)
        Wait( ____ s)

        redFlow = redFlow _______
        blueFlow = blueFlow _______
        }

        Red:    Stop()
        Blue:   Stop()
        Loop Valves_B 5
```

Set the initial flow rate of both syringes

Use a while loop to check when the gradient has fully transitioned to blue

The 'wait' function will wait before moving on to the next line of code

These lines are where you can increment the flow rate values

Note: all flow rates must be greater than 0

Please save your script with a new name when you're done (File>Save As).