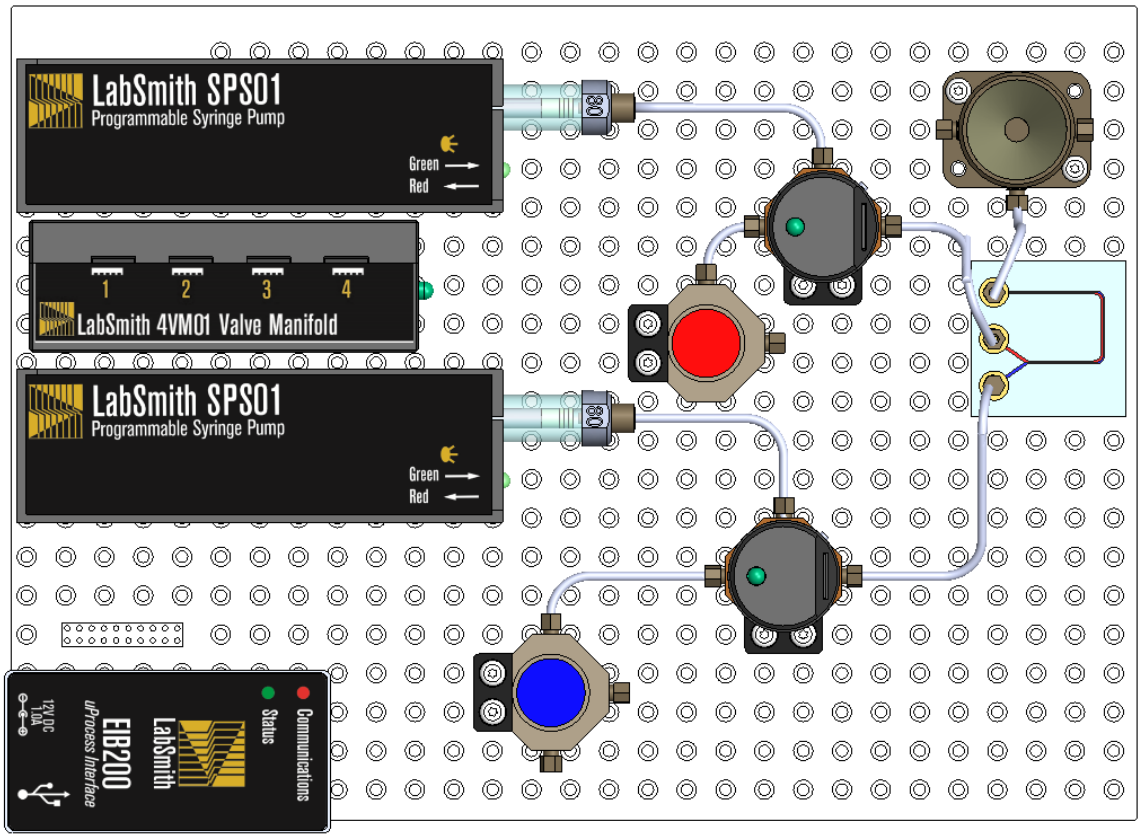
Microfluidic Demonstration

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**

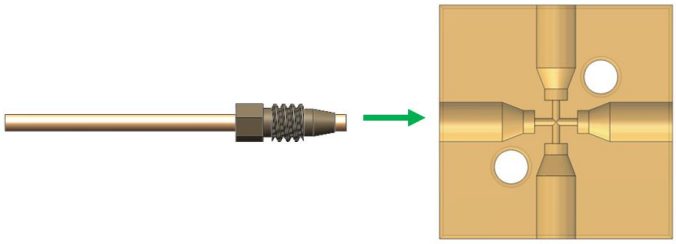
This material is based upon work supported by the National Science Foundation under Grant Number 1621630.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

One-Piece Fitting Assembly Instructions

Insert the tubing into the one-piece fitting. The tubing should protrude through the end of the fitting past the threaded section.

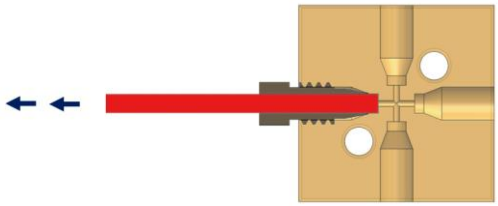
Screw the fitting and tubing the port (see diagram below) until it is hand tight. Ensure that the tubing is securely seated into the port while tightening.



Note: This step is best done by pushing the tubing into the port with one hand and twist the connector in with the other. This step may require some practice to seal the connection.

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

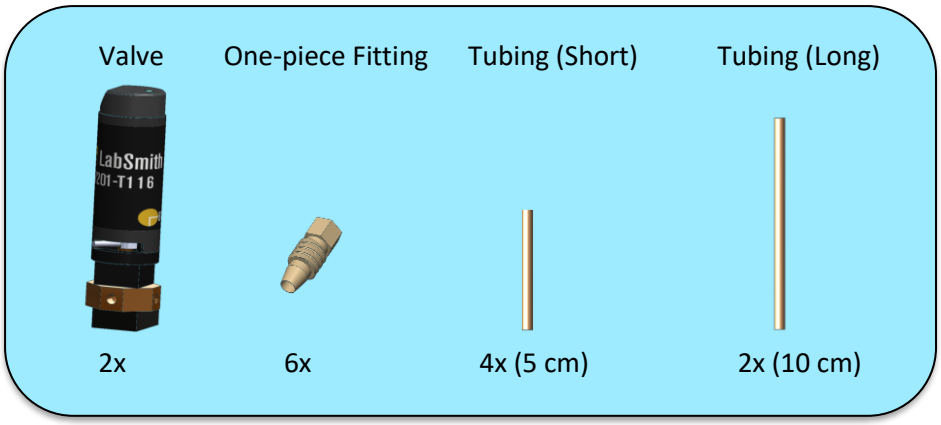
You might have to use the wrench (two provided) to tighten the fitting, but be careful because it is easy to break the fittings with the wrench.



See video at http://labsmith.com/education/demo/before-you-begin/ for more instructions

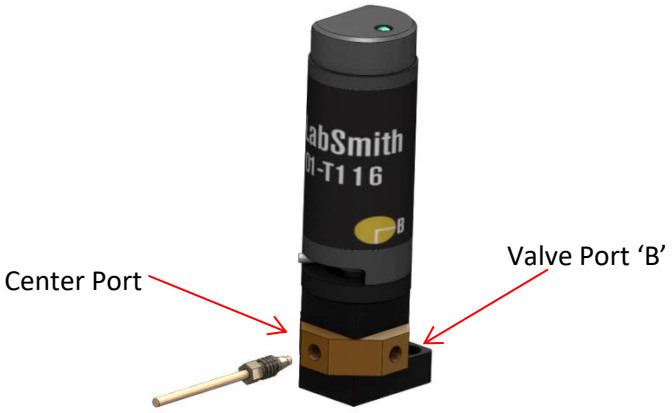
Breadboard Assembly

Step 1: Install Valve Tubing



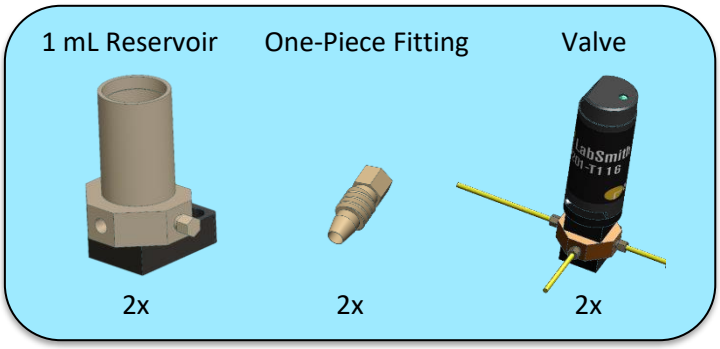
Note: two identical setups will be made in this step. If working in groups, these can be assembled at the same time.

1. With a one-piece fitting, connect a piece of short tubing (5 cm) to the center port of the valve (see diagram below). Do this for each of the two valves**.**



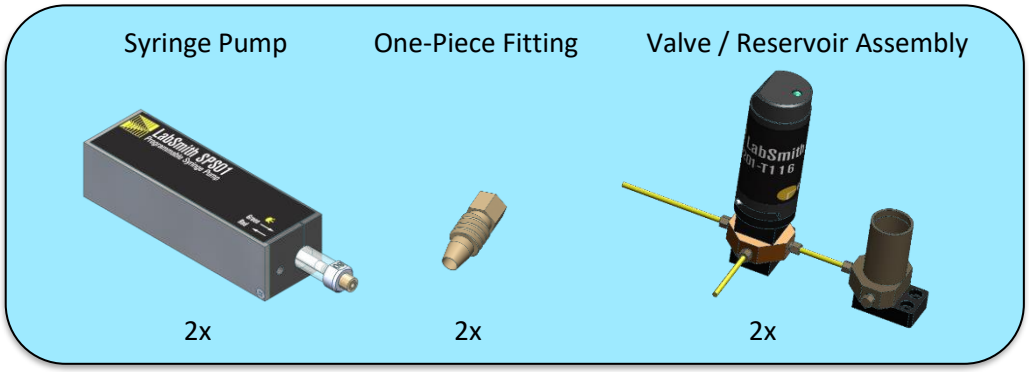
2. Connect a second piece of short tubing to valve port 'B' (see valve label or diagram above). Do this for both valves.
3. Using a piece of 10 cm tubing, secure the tubing to the third port (port A) on the valve. Do this for both valves. When finished, you should have 2 identical valve assemblies.

Step 2: Install Intake Reservoirs



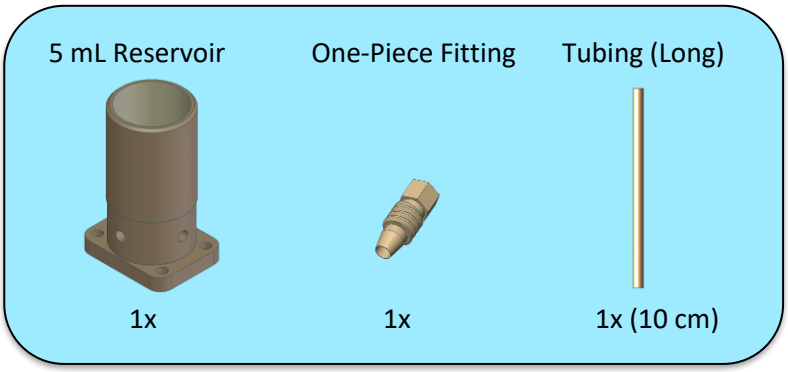
4. Connect tubing from valve port ‘B’ to the open reservoir port using a one-piece fitting. Do this for both valves.

Step 3: Install Syringes

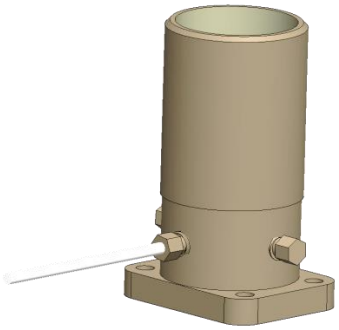


5. Connect tubing from the center valve port to the syringe pump. Complete this for both valve assemblies.

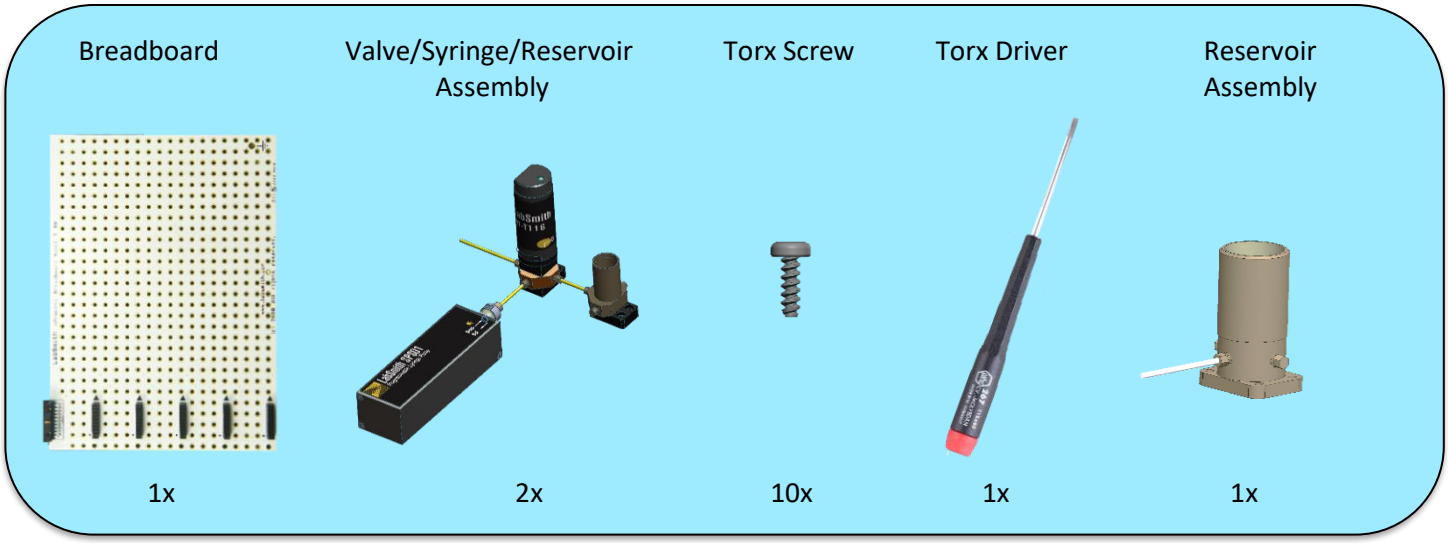
Step 4: Assemble Waste Reservoir



6. Use a piece of 10 cm tubing to connect to the open port on the 5 mL reservoir.

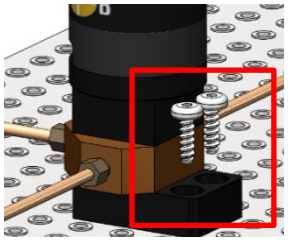


Step 5: Secure the assemblies to a Breadboard

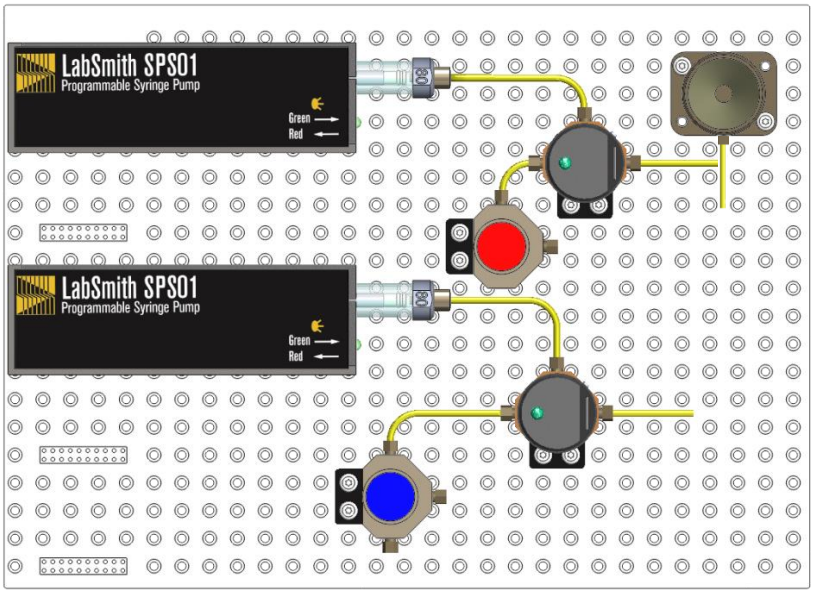


Note: See the diagram below for approximate breadboard layout suggestions.

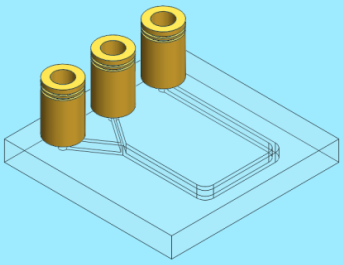
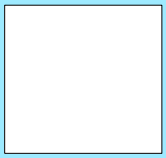
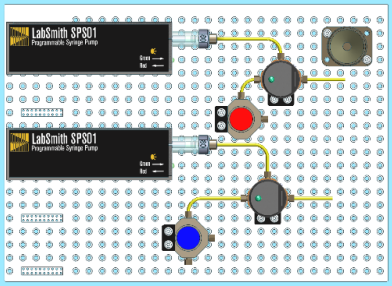
7. Plug a syringe pump (connected to the valve and reservoir) into a connector on the side of the breadboard. (Notice there is a slot on the bottom of the syringe that the connector fits into.)
8. Connect the valve to the breadboard using 2 screws (see diagram below).

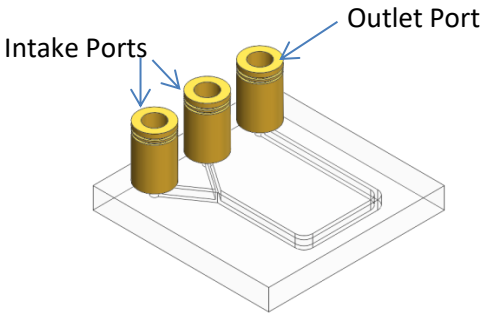


9. Connect the 1 mL reservoir to breadboard using 2 screws.
10. Repeat steps 9-11 for the second valve-syringe-reservoir assembly.
11. Connect the 5 mL Reservoir to the board using 2 screws.

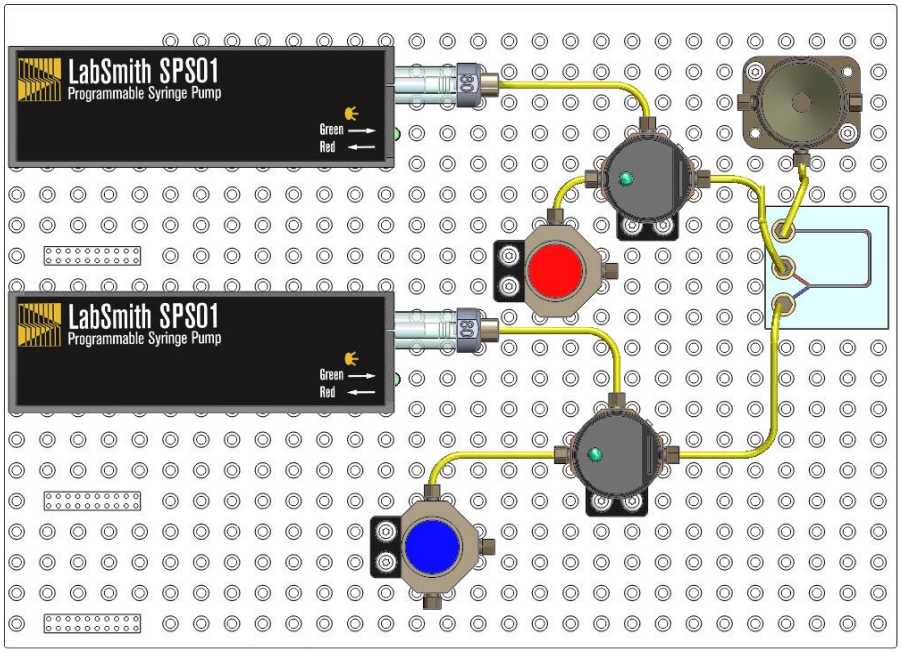


Step 6: Connect the Chip

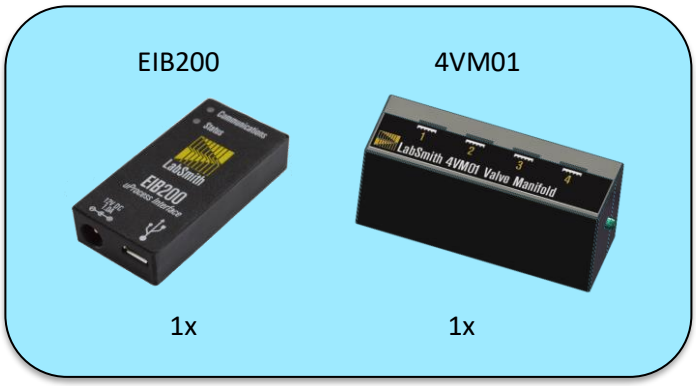
Chip	One-Piece Fitting	Chip Tape	Breadboard Assembly
			
1x	3x	1x	1x



12. Connect the 5 mL Reservoir tubing to the Chip outlet port.
13. Connect the Valve tubing from the 2 valves to the chip's remaining 2 intake ports.
14. Use the double-sided chip tape square to adhere the chip to the breadboard.

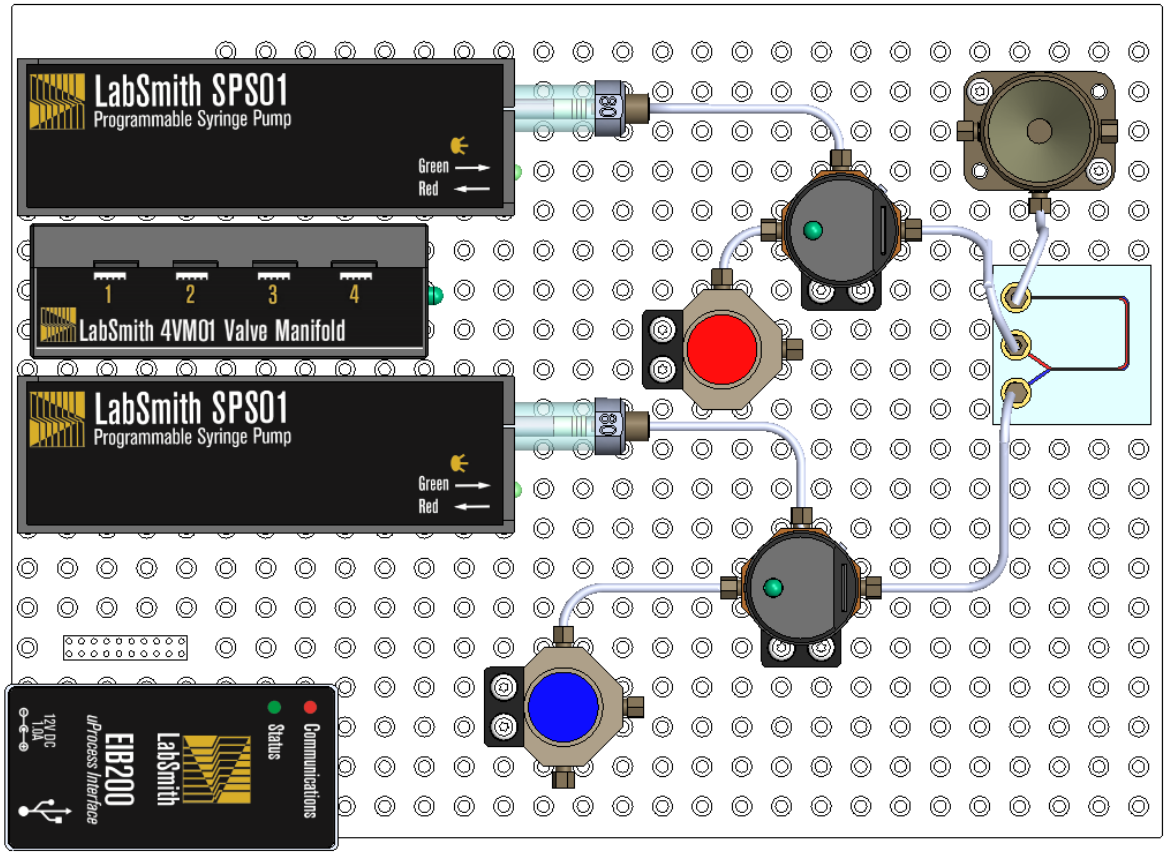


Step 7: Install the EIB200 and 4VM01

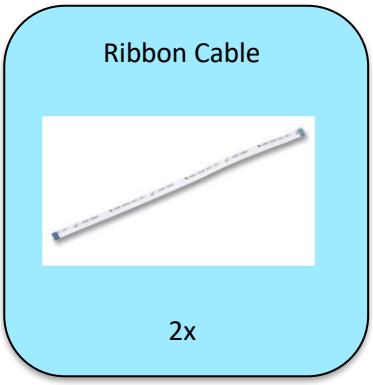


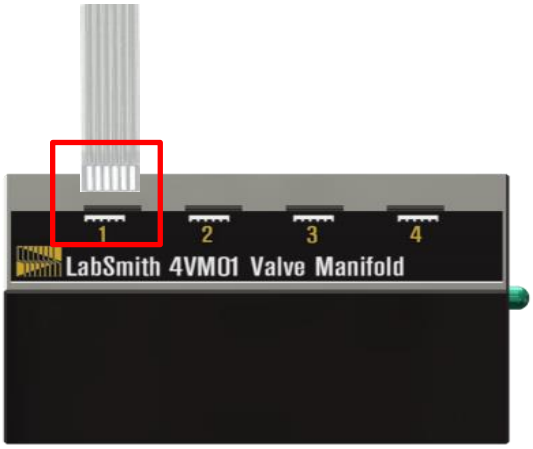
15. Connect the EIB200 and 4VM01 to connectors on the left side of the breadboard.

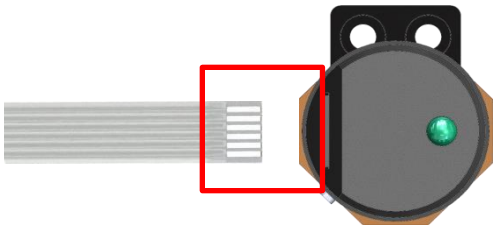
Completed breadboard:



Step 8: Connect Cables







16. Use the two ribbon cables to connect the 4VM01 (valve manifold) to the two valves. Refer to the labels on the 4VM01 and the valves for the correct orientation of the cable. Both ends must be inserted correctly for the valve to function.

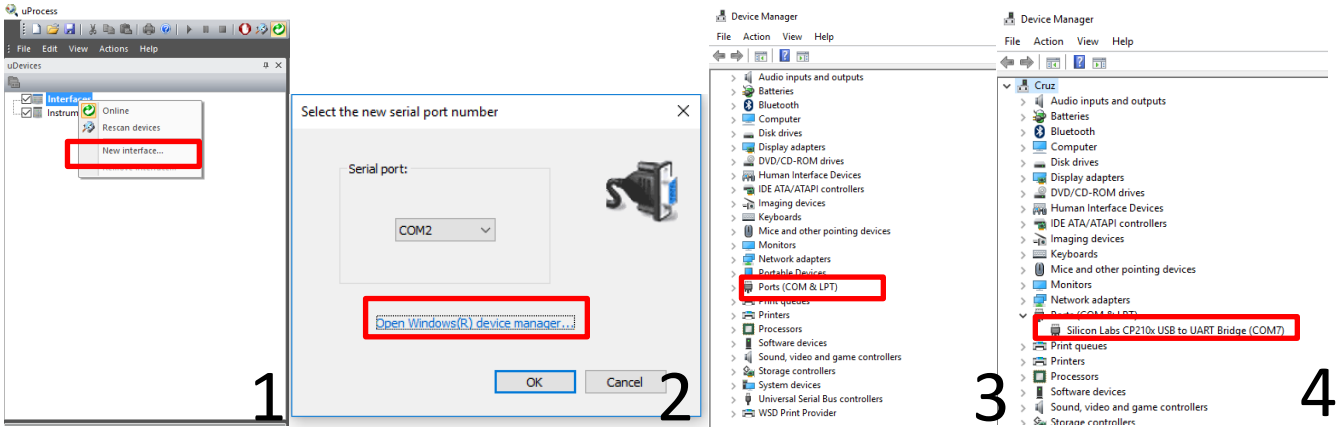
Connect to Software

Step 9: Connect the Breadboard to the Computer

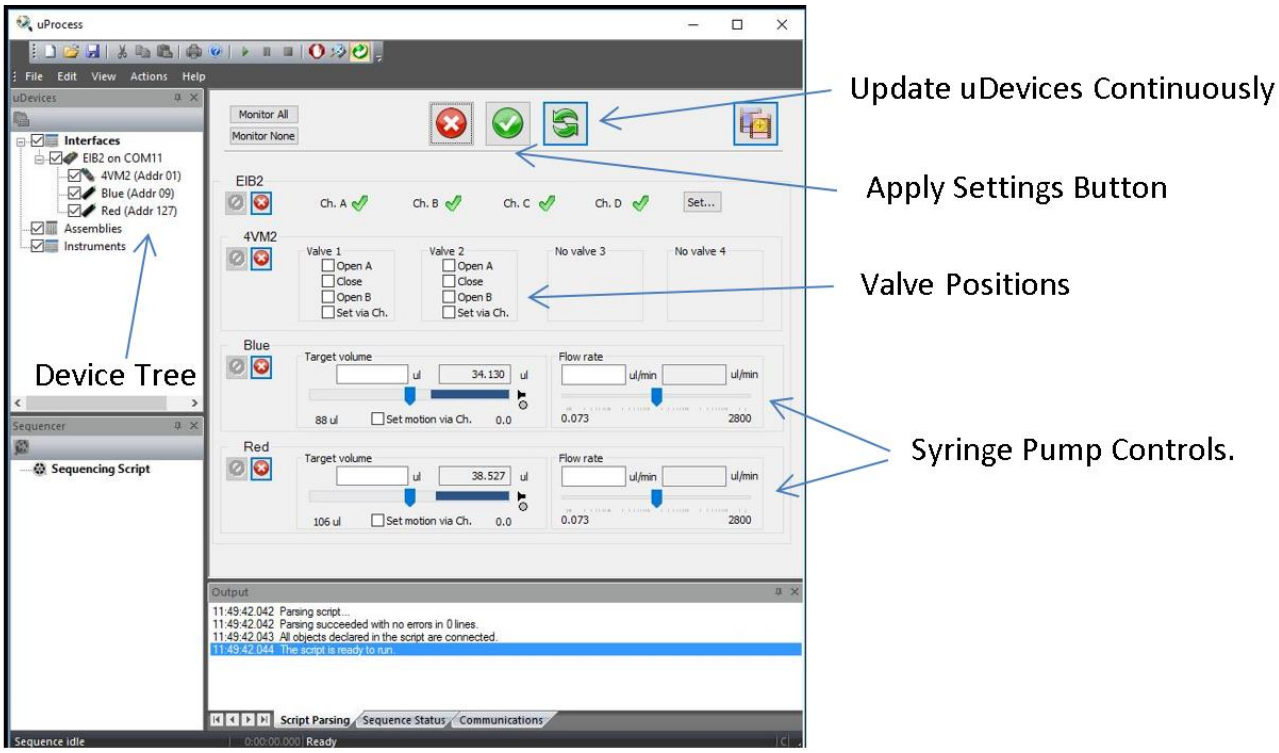
17. Connect the power cable to an outlet and the micro-USB connector into the computer. Plug the power and the micro-USB into the EIB200.
18. Open the uProcess software.

uProcess Icon

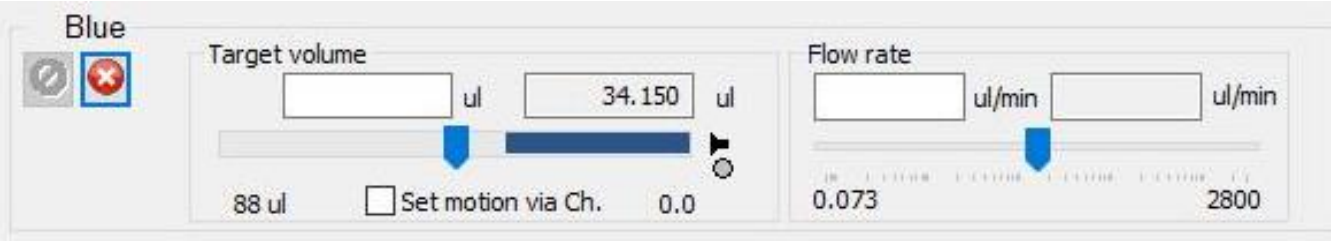
19. Right-click on '*Interfaces*' and select '*New Interface.*' Then select the corresponding COM serial port from the drop-down menu. If the COM port is unknown, click on '*Open Windows device manager…*' and view the Ports list. The correct COM port is listed as '*Silicon Labs CP210x USB to UART Bridge.*'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



Step 10: Manually Control the Syringe Pumps and Valves

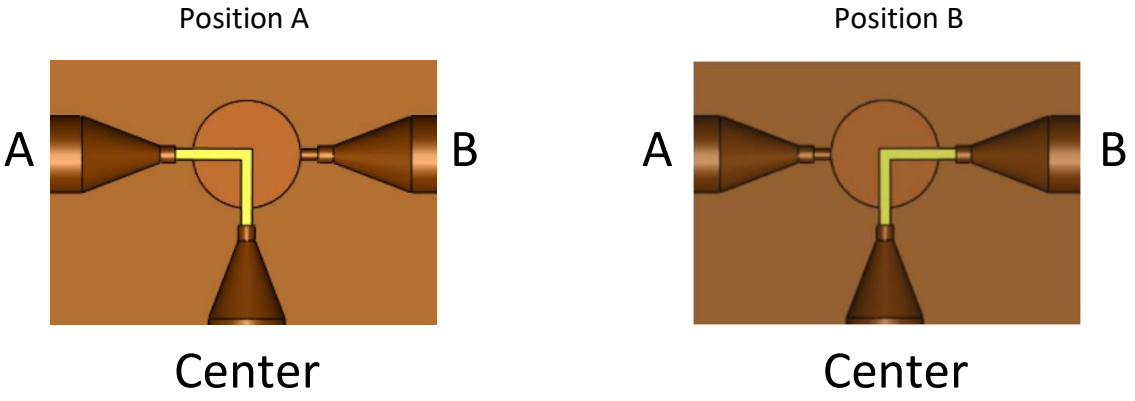


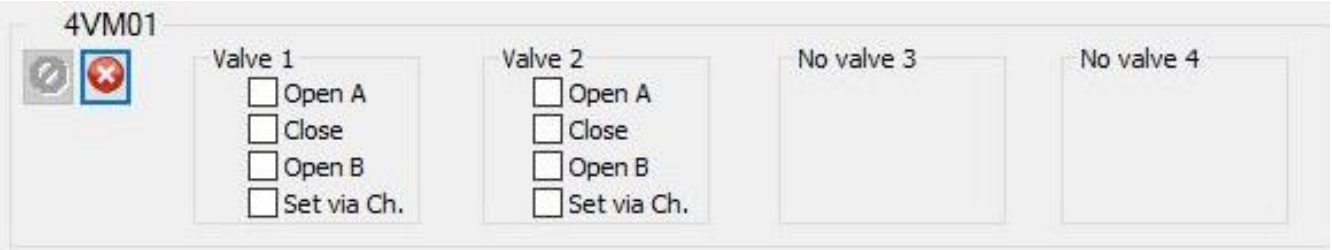
Syringe Pumps: The syringe pumps are used to push or pull the fluids through your microfluidic system. You will notice that the syringe pump has a glass tube with a plunger inside. The software is used to move the plunger back and forth to push and pull the fluids.



20. 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. When the valve is set to position 'A', ports A and Center are connected. When the valve is set to position 'B', ports B and Center are connected.





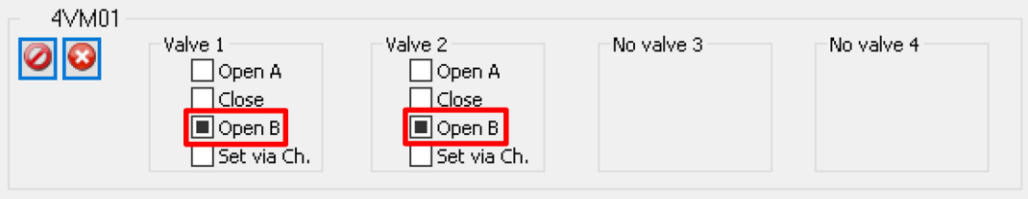
21. Experiment with the valve controls by changing the valve positions and pressing the '*Apply Settings Now*' button ( ) to execute changes. Alternatively, select '*Update uDevices Continuously*' ( ) to apply the settings automatically.
22. Try filling one of the syringes with water from the reservoir. You'll need to set the valve so it is open to the reservoir and then fill the syringe.
23. 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.
24. Now we will automate this process.

Automate your microfluidic controller

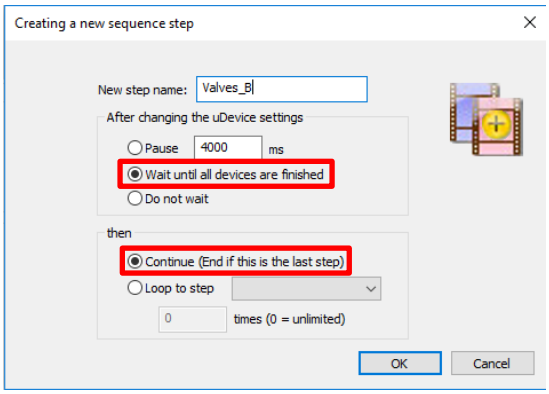
In the next steps you will create an automated script to pump fluids across your microfluidic chip. You will first set the valve position so you can pull liquid from the 1 ml reservoirs to fill the syringes, then change the valve position to push the liquid across the microfluidic chip.

Step 11: Set Valves

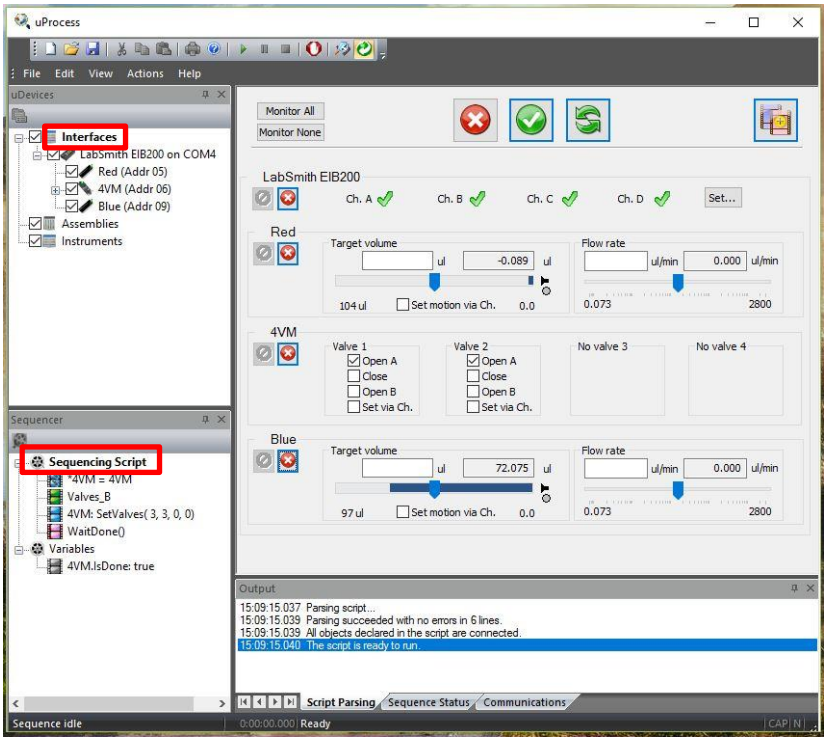
25. Set both valves to position 'B.' This is accomplished by selecting '*Interfaces*' and selecting '*Open B*' from the 4VM01 box (Refer to the screenshot below for details).



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



27. Click on 'Sequencing Script' in the lower left hand corner to open the script editing window.



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

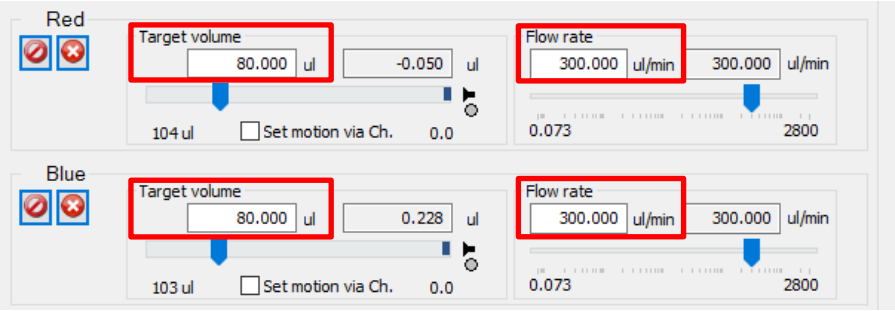
```
*4VM      =          4VM

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 12: Fill Syringes

29. Click on 'Interfaces' in the upper left of your window.
30. 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.
31. 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 (refer to screenshot above).



32. 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()
```

33. Click on 'Interfaces'. Now that you have created code to fill the syringe, you will program the valves and syringe to push the water through the chip.

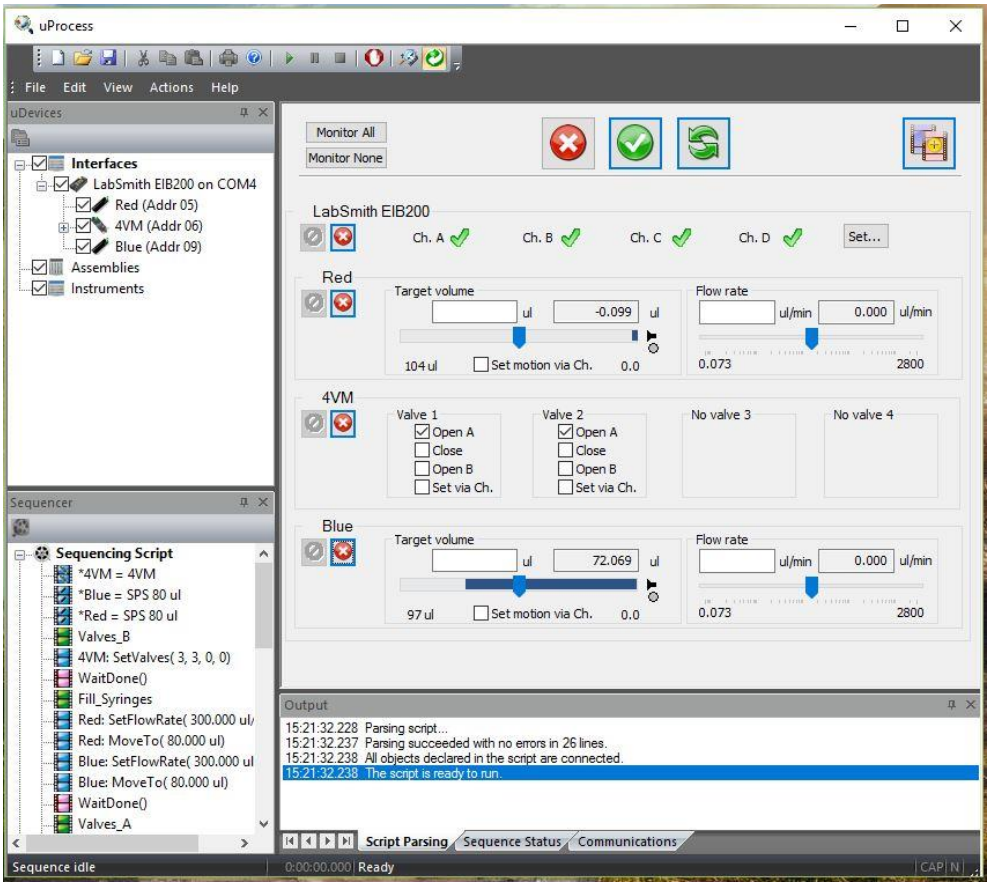
Step 13: Switch Valves

34. 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.
35. Your script will have added the following step:

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

Step 14: Infuse

36. Set both syringe pumps' target volume to 0 ul with a flow rate of 100 ul/min.
37. 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.
38. Your window should now approximate the following:

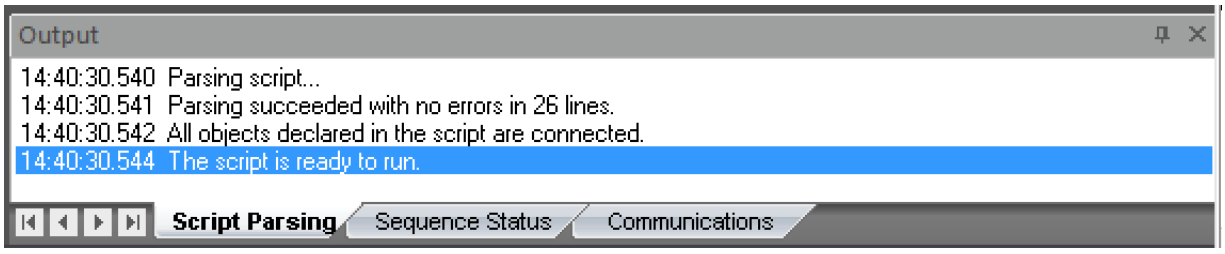


Step 15: Check Script

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

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

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

Infuse:
        Red:     SetFlowRate( 100.000 ul/min)
        Red:     MoveTo( 0.000 ul)
        Blue:    SetFlowRate( 100.000 ul/min)
        Blue:    MoveTo( 0.000 ul)
        WaitDone()
        Loop Valves_B 5
```

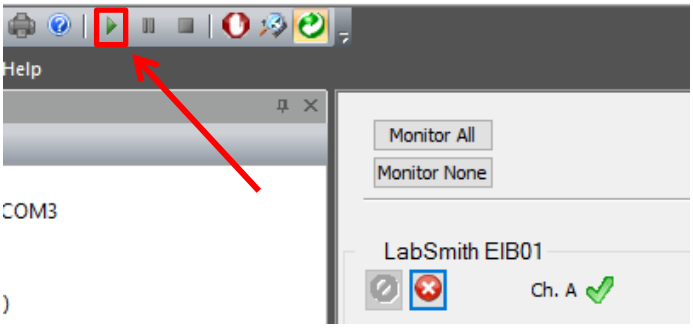
Note: If there are errors in the script it won't run. All errors are described in the output window beneath the script. Double clicking on the error will highlight the line of the error.



Operation

Step 16: Run Automation

40. Fill the two 1 mL reservoirs with water.
41. 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 ( ).

42. The process will run for 5 cycles.
43. Fix any leaks or problems.
44. Once the operation has run a few times and there is no leaking, empty the reservoirs and fill one with red food coloring and the other with blue. (Match the color to the corresponding dot on the syringe pump)
45. 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.
46. After you have run the operation and made observations, click on the script in the lower left corner of the screen.
47. Try changing the flow rate and volume for the red or blue dyes. Run the operation and write down any changes or observations.

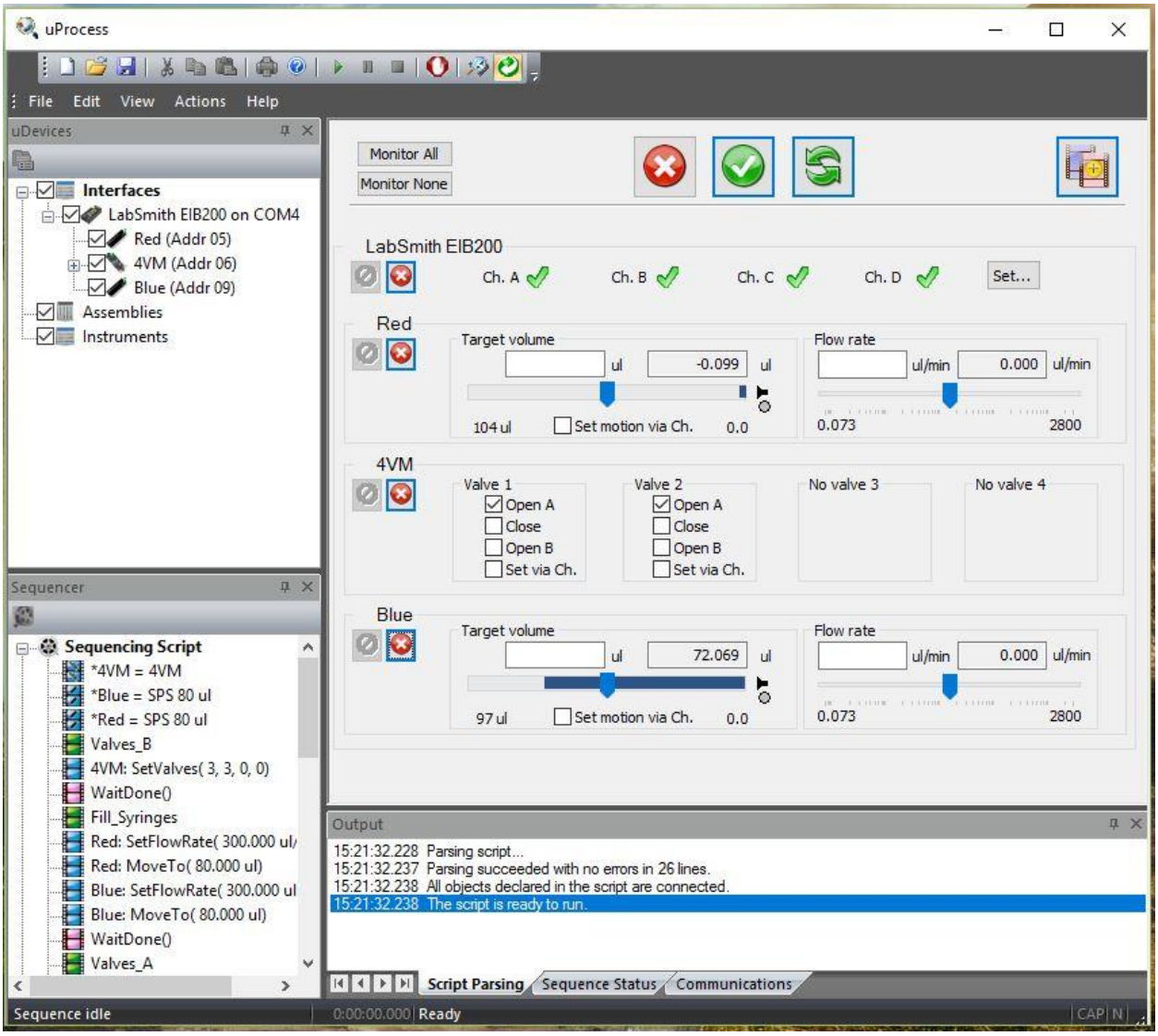
While the script is running, the '*Sequencing Script*' will highlight the current line. Clicking on '*Interfaces*' will show you the status and position of each part.

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

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

Infuse:
        Red:    SetFlowRate( 100.000 ul/min)
        Red:    MoveTo( 0.000 ul)
        Blue:   SetFlowRate( 100.000 ul/min)
        Blue:   MoveTo( 0.000 ul)
        WaitDone()
        Loop Valves_B 5
```

Syringe Set Volume

Dye speed

Flow Manipulation Exercises



Script Commands

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.

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)`

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 **4VM01Name***: SetValves(* **x, x, x, x**)

Sets valve positions.

4VM01Name defined name of 4VM01 (in this case "4VM01")

x, x, x, x position of valves 1, 2, 3, and, 4, respectively, where

x = 0 no change

x = 1: open valve to 'A'

x = 2: close valve

x = 3 open valve to 'B'

Example: `4VM01: SetValves( 1, 0, 3, 2)`

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

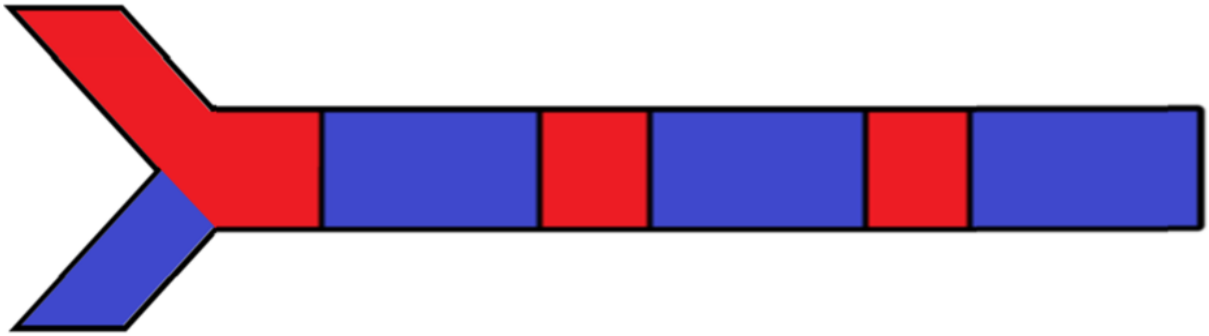
Stops device.

DeviceName defined name of 4VM01 or syringe (must be defined in script header)

Example: `Red: Stop()`

Patterned Flow

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



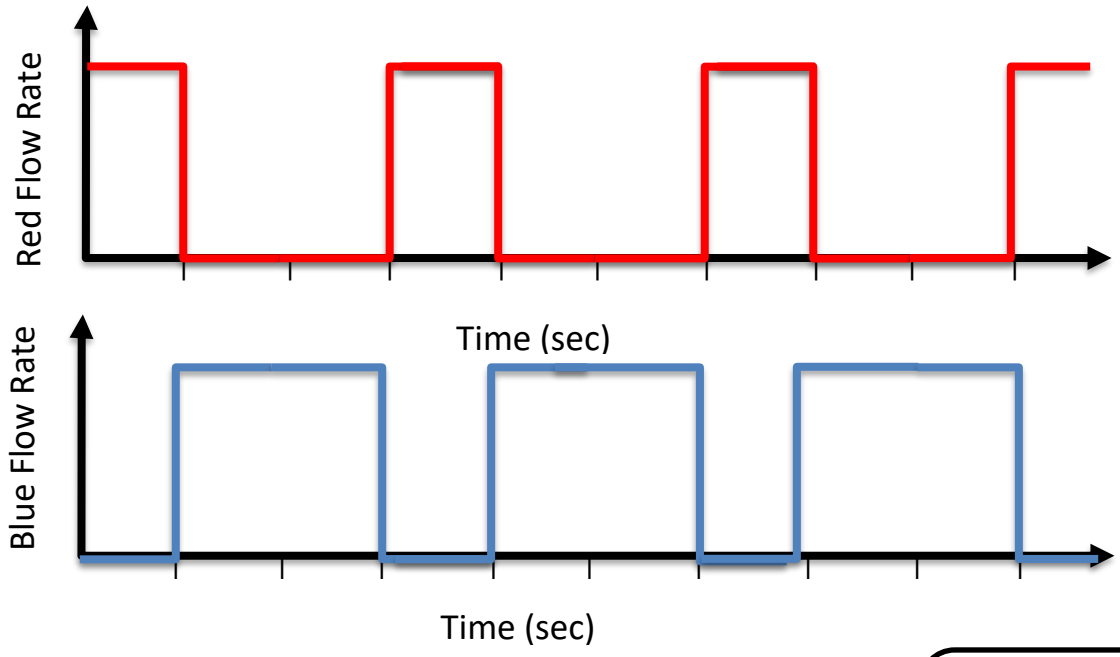
Comprehension Questions:

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. How do you repeat a step a set number of times?

1. Create the flow pattern shown above using the glossary of script functions. The graphs shown below show flow rates and times you could start with.



2. When you have finished, please save your code (File>Save As).

Automation is typically broken into steps. How could you break your code into repeatable

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.

Assigning a variable:

```
a = 10
```

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
```

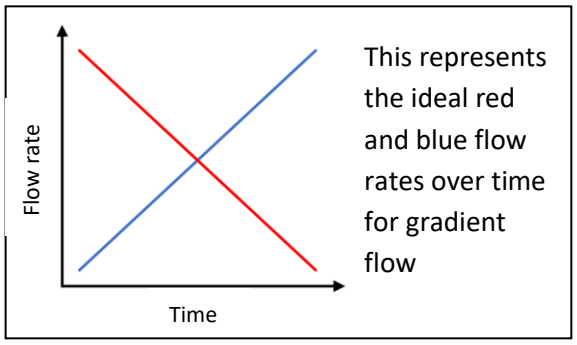
You will use these variables to set flow rates and positions of the syringes. For example, the script below functions the same as the original script you wrote, except it uses variables to assign the flow rate.

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

        a = 100
        R1 = 3 * a
        R2 = a

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

Infuse:
        Red:    SetFlowRate( R2 ul/min)
        Red:    MoveTo( 0.000 ul)
        Blue:   SetFlowRate( R2 ul/min)
        Blue:   MoveTo( 0.000 ul)
        WaitDone()

        Loop Valves_B 5
```

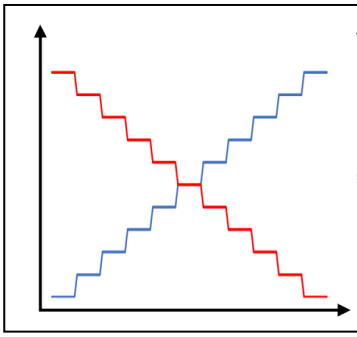
A ‘while’ loop will run the code between the brackets “{ }” while the condition (in this case “a < 10”) is true.

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

You will now create a gradient flow. You want to start out flowing all red dye and then slowly transition to all 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 (0.000 ul)
        Blue:   SetFlowRate (1 ul/min)
        Blue:   MoveTo (0.000 ul)
        Wait( 1 s)

Infuse2
        Red:    SetFlowRate (99 ul/min)
        Red:    MoveTo (0.000 ul)
        Blue:   SetFlowRate (2 ul/min)
        Blue:   MoveTo (0.000 ul)
        Wait( 1 s)
Infuse3
        Red:    SetFlowRate (98 ul/min)
        Red:    MoveTo (0.000 ul)
        Blue:   SetFlowRate (3 ul/min)
        Blue:   MoveTo (0.000 ul)
        Wait( 1 s)

        .
        .
        .
Infuse100
        Red:    SetFlowRate (1 ul/min)
        Red:    MoveTo (0.000 ul)
        Blue:   SetFlowRate (100 ul/min)
        Blue:   MoveTo (0.000 ul)
        Wait( 1 s)
```

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

1. 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 (0.000 ul)
        Blue:   SetFlowRate (blueFlow ul/min)
        Blue:   MoveTo (0.000 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

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