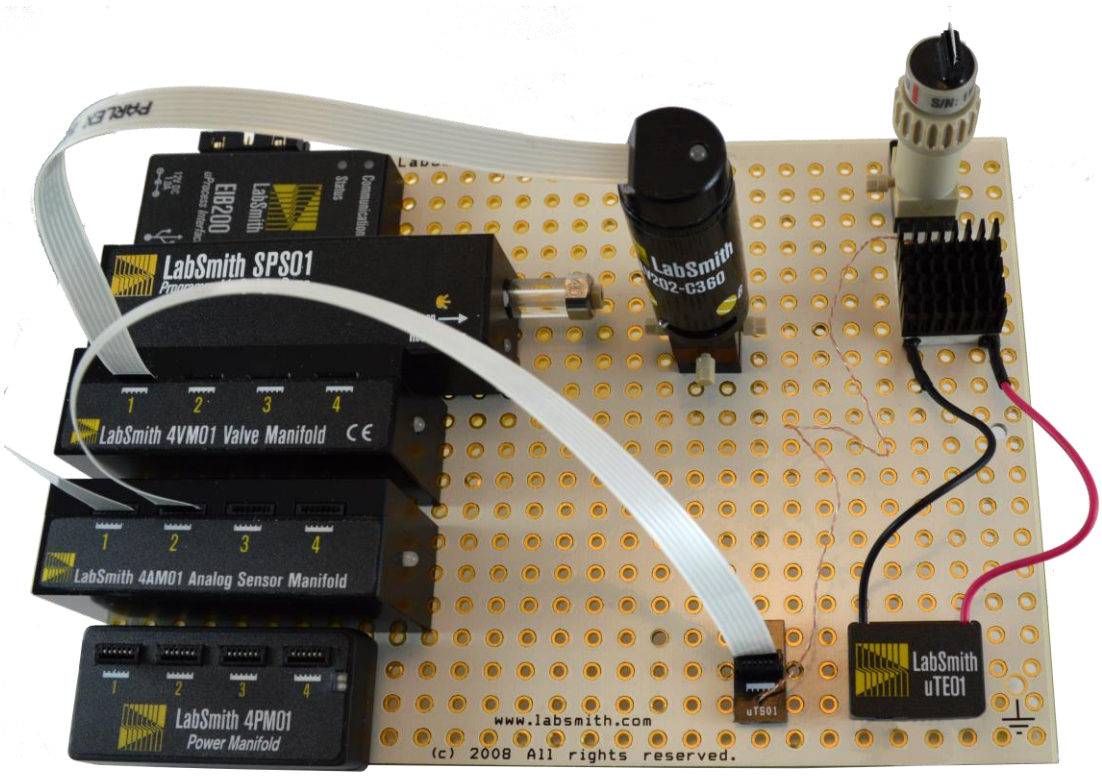
LabSmith uProcess™ System
User Guide

For operation and control of the following
LabSmith uProcess devices using uProcess™ software:

SPS01 Programmable Syringe Pumps
AV-Series Automated Valves
uPS Pressure Sensors
uTE Thermal Modules
uTS Temperature Sensors



6111 Southfront Rd, Suite E ♦ Livermore, CA 94551 ♦ 925-292-5161(ph) ♦ 925-454-9487(fax) ♦ www.labsmith.com

This manual © 2018 LabSmith, Inc. uProcess software © 2018 LabSmith, Inc.
No part of this document may be reproduced or distributed without the consent of LabSmith.

LabSmith, CapTite, uProcess and uDevice are trademarks of LabSmith, Inc.

CHAPTER 1. INTRODUCTION 4

Equipment List 5
Cautions and Warnings 6
Getting Help 6
About This Manual 6
Converting Scripts from Version 1.X to 2.X 7

CHAPTER 2. GETTING STARTED 9

Connecting an Electronic Interface Board 9
Adding a Breadboard 9
Installing uDevices 10

CHAPTER 3. uPROCESS SOFTWARE 12

Installing uProcess Software 12
Software Interface 12
Communicating with uDevices 14
Online and Offline 18
Device Naming and Numbering 18
Saving Data into a Log File 19
External Instruments 20

CHAPTER 4. REAL TIME DEVICE CONTROL 21

4AM 22
Analog Manifold Control 22
4PM 27
Power Manifold Control 27
SPS Syringe Pump Control 32
4VM Valve Manifold Control 34

CHAPTER 5. AUTOMATED CONTROL 37

Opening Script Files 37
Create a Script Using the Wizard 37
Save a Script File 39
Run, Pause and Quit a Script 39
Automated Script Functions 40
General Formatting Rules 46
Editing Automated Script Text 51
Example Scripts 54
Thermal Application Example: Switching Between Two Set Temperatures 65

CHAPTER 6. SVM340 AUTOMATION AND CONTROL 69

APPENDIX A. MENUS, TOOLBARS, WINDOW CONTROLS 76

APPENDIX B. MAINTENANCE 79

Updating Firmware 79
Valve Actuation Profile Settings 79
Changing the Syringe Pump Glass Assembly 80
Calibrating the Syringe Pump 81
Care of the Syringe Glass and Plunger 82

APPENDIX C. uDEVICE CURRENT CONSUMPTION 84

APPENDIX D. CAPILLARY AND TUBING INSTALLATION GUIDE 85

CHAPTER 1. INTRODUCTION

LabSmith's uProcess™ is a platform comprised of a suite of hardware and software products that simplify construction and control of microfluidic systems.

uProcess automated devices work together with CapTite™ microfluidic interconnect products to make it easy to build and rebuild micro- and nano-fluidic circuits. All products are inter-compatible and quick to assemble, taking the hassle out of fluid routing.

- **SPS01 Programmable Syringe Pumps** deliver volumes up to 100 microliters, with resolution as low as 2 nanoliters. The pump body design allows syringes to be easily removed and replaced, providing a wide range of volumes with a single pump.
- **AV-Series Automated Valves** with zero dead volumes and low swept volumes. The AV-Series valve are controlled with a **4VM01** or **4VM02** four-channel valve manifold.
    - **AV201**: 2-position, 3-port switching valve
    - **AV202:** 2-position, 4-port switching valve
    - **AV303:** 2-position, 6-port switching valve
    - **AV801**: 8-position, 9-port selector valve
- **uPS Pressure Sensors** provide pressure sensing and regulation. The uPS sensors are controlled with a **4AM01** four-channel sensor manifold.
- **uTS Temperature Sensors** provide temperature sensing and regulation. The uTS sensors are controlled with a **4AM01** four-channel sensor manifold.
- **uTE Thermal Controllers** provide thermal control (heating and cooling). The uTE01 and uTE02 controllers are controlled with a **4PM01** four-channel power manifold.

The **uProcess Software** provides a simple interface for configuring and controlling uProcess devices and for creating automated processes for coordinated control.

This user guide will show you how to connect, configure and control uProcess devices, how to install and operate uProcess software, and how to create automated flow control processes.

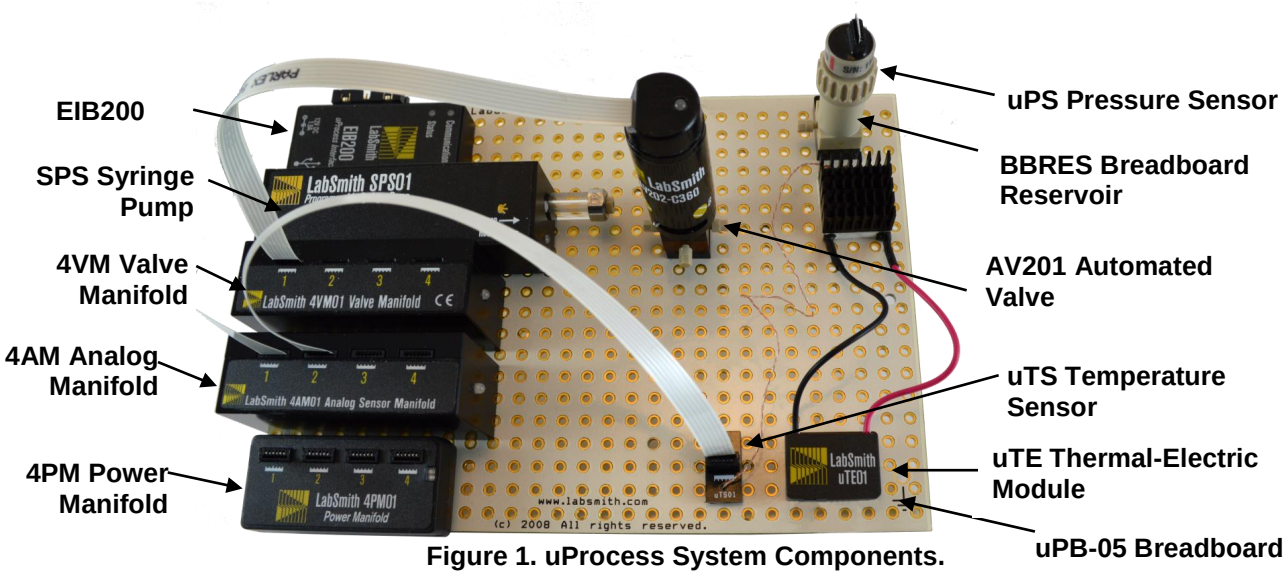


Figure 1. uProcess System Components.

Equipment List

Required Equipment

- PC-compatible computer running Windows® 7 or newer
- uProcess Interface (EIB200)
- uProcess™ software - installation disk included with EIB purchase
- 12V power supply for EIB – included with EIB200 purchase
- micro-USB cable – included with EIB200 purchase

Additional required and optional equipment for **syringe pumps**

- SPS01 Programmable Syringe Pump
- SPS01 syringe glass/plunger set – included with SPS01 purchase
- CapTite ferrule (C360-100, T132-100, or T116-100) for connecting to capillary or tubing
- syringe glass/plunger installation tools (plunger wrench and 3/32” hex wrench)
- installation screws (qty 2 x M2-0.4), optional

Additional required and optional equipment for **automated valves**

- AV201, AV202, or AV303, or AV801 automated valve
- 4VM01 or 4VM02 valve manifold (one manifold per up to four valves)
- flat flex cable (one per valve) – 4 cables included with 4VM01 or 4VM02
- CapTite ferrules (C360-100, T132-100, or T116-100) for connecting to capillary or tubing
- breadboard mounting screws (2-28x1/4”), two per valve, optional

Additional equipment required for **pressure sensors**

- uPS pressure sensor
- 4AM01 analog manifold (one manifold per up to four sensors)
- flat flex cable (one per sensor) –4 cables included with 4AM01
- CapTite component (-C360, -T132, or –T116) with female port for installing sensor

Additional equipment required for **temperature sensors**

- uTS temperature sensor
- 4AM01 analog manifold (one manifold per up to four sensors)
- flat flex cable (one per sensor) –4 cables included with 4AM01
- breadboard mounting screw (2-28x1/4”), one per sensor, optional

Additional equipment required for **temperature controllers**

- uTE01 or uTE02 temperature controller
- 4PM01 power manifold (one manifold per up to four uTE controllers)
- flat flex cable (one per uTE) – 4 cables included with 4PM01
- breadboard mounting screw: 2 x M2-0.4 for uTE01, 2 x 2-28x3/8” for uTE02, optional

- heat sink: uTE controllers should be attached to a heat sink on one or both sides of the Peltier to minimize chance of overheating
- temperature sensor: one or more temperature sensors should be used with each uTE controller to ensure the controller temperature stays within the operating range.

Optional Equipment

- 20-pin communication cable for direct connection of EIB to uDevice (included with EIB200 purchase)
- uPB or iBB breadboard

Cautions and Warnings

When working with uProcess devices and accompanying electronics, follow standard procedures to protect your devices from excessive static electricity, power fluctuations, etc.

Read and understand all documentation for your individual uProcess devices and components before connecting the equipment.

CAUTION: Extra care must be taken when using the electronic components with fluids. Breadboards are designed with the electronics mounted on one side; use precautions when introducing liquids to this side of the breadboard.

Getting Help

This guide is your main source for information on operating uProcess software. Choose **Help>About uProcess**, then click the **Open User Manual** link to browse this manual electronically, in Adobe® Acrobat® PDF format.

The LabSmith web site (***www.labsmith.com***) also includes helpful applications notes and other technical information to help you use uProcess software and devices. If you are unable to find the information you need, please contact our technical support team at (925) 292-5161, or email us at ***support@labsmith.com***. Be prepared to give the software version. To determine the software version number, select **Help>About uProcess**.

About This Manual

Several conventions are used throughout this manual to highlight important concepts:

Menu>Option Menu bar selections will be shown in bold text, with descending options separated by the ">" symbol. For example, **Help>About uProcess** indicates that you should click the **Help** menu, and then click **About uProcess**.

Note Notes indicate helpful tips or additional information.

CAUTION These notes signify important considerations that will help you to avoid losing data or damaging hardware.

Italics When browsing the electronic version of this manual, click on these italic cross-references to view other pertinent sections of the manual.

Choose **Help>About uProcess**, then click the **Open User Manual** link to browse the electronic version of this manual.

Please contact LabSmith or visit ***www.labsmith.com*** for user manual, software and firmware updates.

Converting Scripts from Version 1.X to 2.X

Version 2 of the uProcess software includes many new scripting features, such as variable and math support. Inclusion of these features required changes to many of the script commands from uProcess V1. The table below gives an overview of changes from uProcess V1 to V2, see **Chapter 5** for the complete list of uProcess script commands.

Commands uProcess 1.0	**Commands uProcess 2.0**
<Beep>	**Beep()**
<Break>	**Break**
<Goto> StepName	**Goto** StepName
DeviceName: **HysteresisReg** Pmin Pmax units *DeviceName*: **HysteresisReg** Pmin Pmax units to **Ch**X SyringeName: **MoveWithCh**X (or **MoveOppositeCh**X)	*DeviceName*: **HysteresisReg**(Pmin units, Pmax units) Ex: `HysteresisReg(40 kPa, 60 kPa)` *DeviceName*: **HysteresisRegTo**(Pmin units, Pmax units) SyringeName: **MoveWith(DCh**X) (or **MoveOpposite(DCh**X))
DeviceName: **IfDone** *DeviceName*: **IfNotDone**	*DeviceName*: **IfDone()** *DeviceName*: **IfNotDone()**
<log> on *optional filename* **<log> off**	**log(on)** **or** **log(***optional filename***)** **log(off)**
<Loop> *StepName cycles*	**Loop** *StepName cycles*
SyringeName: **MoveTo** *vol units*	*SyringeName*: **MoveTo**(*vol units*)
<Pause> *duration units*	**Wait(***duration units***)**
SyringeName: **SetFlowRate** *rate units*	*SyringeName*: **SetFlowRate(***rate units***)**
*4VM_Name***: SetValves** *x x x x*	*4VM_Name***: SetValves** (*x, x, x ,x*)
DeviceName: **Stop**	*DeviceName*: **Stop()**
<Quit>	**Quit**
SensorName: **RegDownTo** *value units* *SensorName:* **RegUpTo** *value units*	*SensorName:* **RegDownTo** *(value units)* *SensorName:* **RegUpTo** *(value units)*
SensorName: **RegDownTo** *value units* ***to ChX*** *SyringeName:* **MoveWith Ch***X* *(or SyringeName:* **MoveOpposite Ch***X*)	*SensorName*: **RegDownTo(***value units, ***DChX)*** (or **RegUpTo(***value units, ***DChX))*** *SyringeName:* **MoveWith(DCh***X)* *(or SyringeName:* **MoveOpposite(DCh***X*)
SensorName: **RegOff**	*SensorName*: **RegOff()**
<WaitAllDone>	**WaitDone**()
*DeviceName***: WaitDone**	*DeviceName:* **WaitDone()**

uProcess version 2.0 also supports several new commands including SetChannels, SetCurrent,SetOff, SetPerChannel, and SetPower. See the command descriptions in **Automated Script Functions** for more details.

Generic Formatting Updates

- Step names must be followed by a colon (":")
- Device names and step names cannot contain a dash ("-") (see **Device Naming and Numbering** for instructions on changing device names).

New Functionality: Variables, If Statements, and While loops

Variables can be defined and operated on in the script. Properties of connected devices can also be accessed in the script while it is running.

```
x1 = 20                          ; declare variables
Voltage = sqrt(45) + sin(5*t) ; use mathematical expressions
position = Syringe.Volume     ; access variables inherant to each device
If (uPS.Reading >=50)          ; use an if statement
      SPS01:      SetFlowRate (SPS01.MaxFlowRate ul/min)
While (a <=10)                 ; perform a while loop
{
      Beep()
      Wait(1 s)
      a = a + 1
}
```

More information on these functions and further uses and functions can be found in **Chapter 5**.

uProcess Version 1.x

```
*Manifold       =       4VM
*Syringe        =       SPS     80 ul

Open_Reservoir
        Manifold:SetValves 1 0 0 0
        <WaitAllDone>

Fill_Syringe
        Syringe:SetFlowRate 100.000 ul/min
        Syringe:MoveTo 80.000 ul
        <WaitAllDone>

Open_Process
        Manifold:SetValves 3 3 0 0
        <WaitAllDone>

Infuse
        Syringe:SetFlowRate 10.000 ul/min
        Syringe:MoveTo 0.000 ul
        <WaitAllDone>
        <Loop> Fill_Syringe 5
```

uProcess Version 2.x

```
*Manifold       =       4VM
*Syringe        =       SPS     80 ul

Open_Reservoir:
        Manifold:SetValves( 1, 0, 0, 0)
        WaitDone()

Fill_Syringe:
        Syringe:SetFlowRate( 100.000 ul/min)
        Syringe:MoveTo( 80.000 ul)
        WaitDone()

Open_Process:
        Manifold:SetValves( 3, 3, 0, 0)
        WaitDone()

Infuse:
        Syringe:SetFlowRate( 10.000 ul/min)
        Syringe:MoveTo( 0.000 ul)
        WaitDone()
        Loop Fill_Syringe 5
```

CHAPTER 2. GETTING STARTED

Connecting an Electronic Interface Board

Though the devices you use with your uProcess system will vary, some basic hardware is common to all uProcess setups.

Every system requires at least one uProcess Interface (EIB200), which can control up to 110 SPS01 Syringe Pumps, 4VM02 Valve Manifolds, 4PM01 Power Manifolds, and/or 4AM01 Analog Manifolds. Since each manifold can control up to four devices, one EIB200 can thereby control hundreds of devices.

Although over 100 devices can be sequentially controlled using a single EIB200, the number of devices simulaneously actuated is limited to approximately 10, based on the current capacity of the EIB200 power supply. See Appendix C for information to calculate the current consumption for a particular setup. Multiple EIB200s can be used at one time when a large number of devices or concurrent operations are required.

To connect an EIB200 to the computer:

- Use the included mini-USB cable to connect the COM port on the EIB200 to the computer's USB port.
- Connect the included power transformer from the EIB200 to a wall outlet. The two indicator lights will flash when powered up, after which the Power light will stay illuminated red. The COM light will illuminate when the EIB is communicating with the computer.

CAUTION: Connecting or disconnecting uDevices while the EIB200 is powered (hot swapping) can damage the uDevices. We recommend unplugging the EIB200 power connector before uDevices are connected or disconnected.

Adding a Breadboard

An EIB200 can be connected to a single syringe pump, valve manifold, analog manifold, or power manifold; however, it is typically used with one of the LabSmith breadboards (uPB-5, uPB-8, uPB-12, or iBB) for mounting multiple components (***Figure 2***). Between 5 and 12 uDevices (syringe pumps, valve manifolds, analog manifolds, power manifolds) can be connected to a single breadboard.

The EIB200 is connected to the uDevices in one of the following ways:

1. Plugged directly into a slot on a uProcess breadboard (uPB-05, uPB-08-uPB-12,or iBB)
2. Connected to uProcess breadboard via a communication cable.
3. Connected directly to a uDevice (SPS01, 4AM01, or 4VM02) via a communication cable. Note: the standard communication cable provided with the EIB200 can connect only 1 uDevice. Contact LabSmith about obtaining a custom communication cable if connecting multiple uDevices without a breadboard.

The larger breadboards (uPB-8, uPB-12, and iBB) can be daisy-chained together by installing a 20-pin communication cable between female connectors on each board.

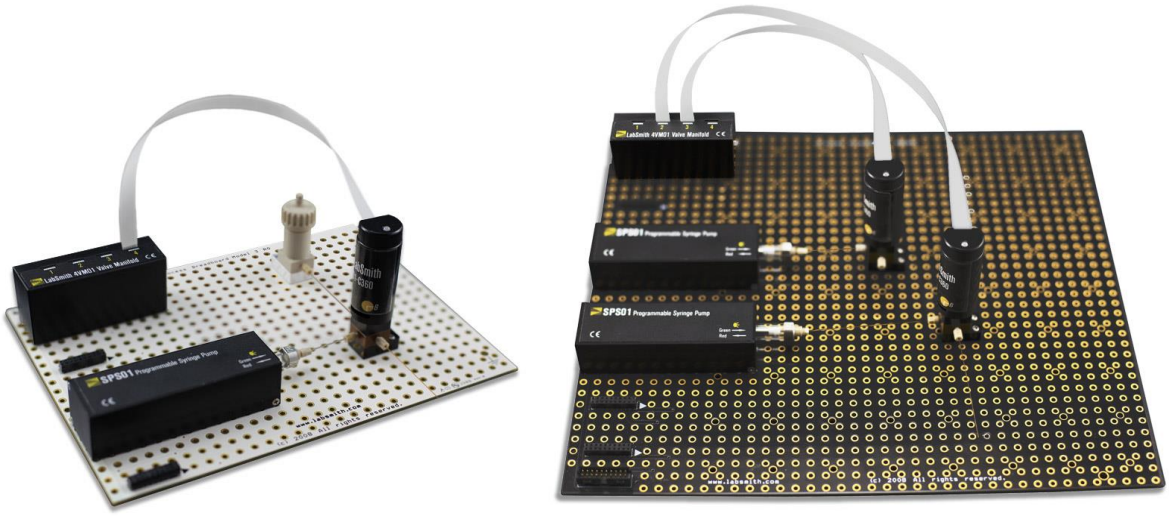


Figure 2. uPB-05 and uPB-08 Breadboards with uDevices.

Installing uDevices

The following sections provide information on installing the various devices available for the uProcess system.

Note: disconnect power to the EIB200 before adding or removing devices

Installing Automated Valves

To install and configure AV201, AV202, AV303, or AV801 automated valves:

- Plug a 4VM01 or 4VM02 valve manifold into one of the connectors located along the side of the breadboard.
- Secure the automated valve(s) to the desired location on the breadboard using two ¼" breadboard mounting screws (optional).
- Use a ribbon cable to connect the automated valve to a port on the valve manifold. Observe the label on the valve and manifold to ensure the cable is connected at the correct orientation.
- Upon power up, the LED on the 4VM01 /4VM02 will flash red/green and then glow steady green. The valve LED does not illuminate during power up.

NOTE: for uProcess Software Version 2.0+, the 4VM must be configured with the valve type (AV201, AV202, AV303, or AV801) for proper operation. Refer to Chapter 4 for instructions on configuring the 4VM.

Installing Syringe Pumps

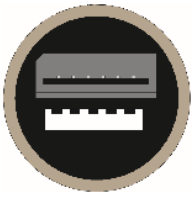
To install an SPS01 programmable syringe pump, plug the syringe pump body into one of the connectors located along the side of the breadboard. Two M3 screws may optionally be used to more securely attach the syringe to the bottom of the breadboard.

Upon power up, the LED on the SPS01 will flash red/green and then go out.

Installing Pressure or Temperature Sensors

To install and configure a uPS pressure sensor or uTS temperature sensor:

- Plug the 4AM01 sensor manifold into one of the connectors located along the side of the breadboard.
- Pressure sensors:
    - Secure the pressure sensor(s) to the desired component. Note: the uPS-360 sensors can be installed in a –C360 or –T132 CapTite component (for use with 360µm OD tubing or 1/32" OD tubing, respectively) The uPS116 sensors can be installed in a –T116 component, for use with 1/16" OD tubing.
    - Use a ribbon cable to connect the pressure sensor to a port on the analog manifold. Observe the label on the manifold to ensure the cable is connected at the correct orientation.
    - Connect the ribbon cable to the uPS sensor with the following orientation:



- Temperature sensors:
    - uTS01-STD sensors can be attached to the substrate via thermal tape or epoxy.
    - uTS01-INS sensors can be immersed in liquid (such as via a reservoir).
    - Connect a **ribbon cable** from each temperature sensor to a port on the 4AM01. Labels on the uTS and 4AM indicate the correct orientation of the cable.
    - Optional: secure each uTS01 sensor body to the breadboard using a 2-28-¼" screw.

Upon power up, the LED on the 4AM01 will flash red/green and then turn off.

Installing uTE Thermal Controllers

To install and configure a uTE01 or uTE02 thermal controller:

- Plug a **4PM power manifold** into one of the connectors located along the left side of the breadboard.
- Connect a **ribbon cable** from each uTE to a port on the 4PM. Labels on the 4PM and uTE02 indicate the correct orientation of the cable. Refer to the figure below for the correct orientation of the cable on the uTE01.

NOTE: Each channel of the 4PM must be configured with the load type (i.e. uTE08 or uTE12) for proper operation. Refer to Chapter 4 for instructions on configuring the 4PM.

CHAPTER 3. uPROCESS SOFTWARE

Installing uProcess Software

uProcess software runs on any PC-compatible computer running Microsoft Windows® 7, 8, or 10. The software requires approximately 5MB of free disk space. To install uProcess:

- Insert the uProcess for Windows CD.
- Double-click **SETUP.EXE** and follow the on-screen instructions.
- To run the software, double-click the **uProcess** icon on your desktop, or choose **Run>uProcess>uProcess**.

The uProcess software can also be downloaded from the LabSmith website (www.labsmith.com) . We recommend visiting the website occasionally to check for software updates.

Software Interface

The **uProcess Dashboard** consists of a menu bar, toolbar, and configurable windows (***Figure 3***). Unique menu and toolbar commands will be described in the body of the text below. A complete listing of all menu and toolbar commands can be found in ***Appendix A. Menus, Toolbars, Window Controls*****.**

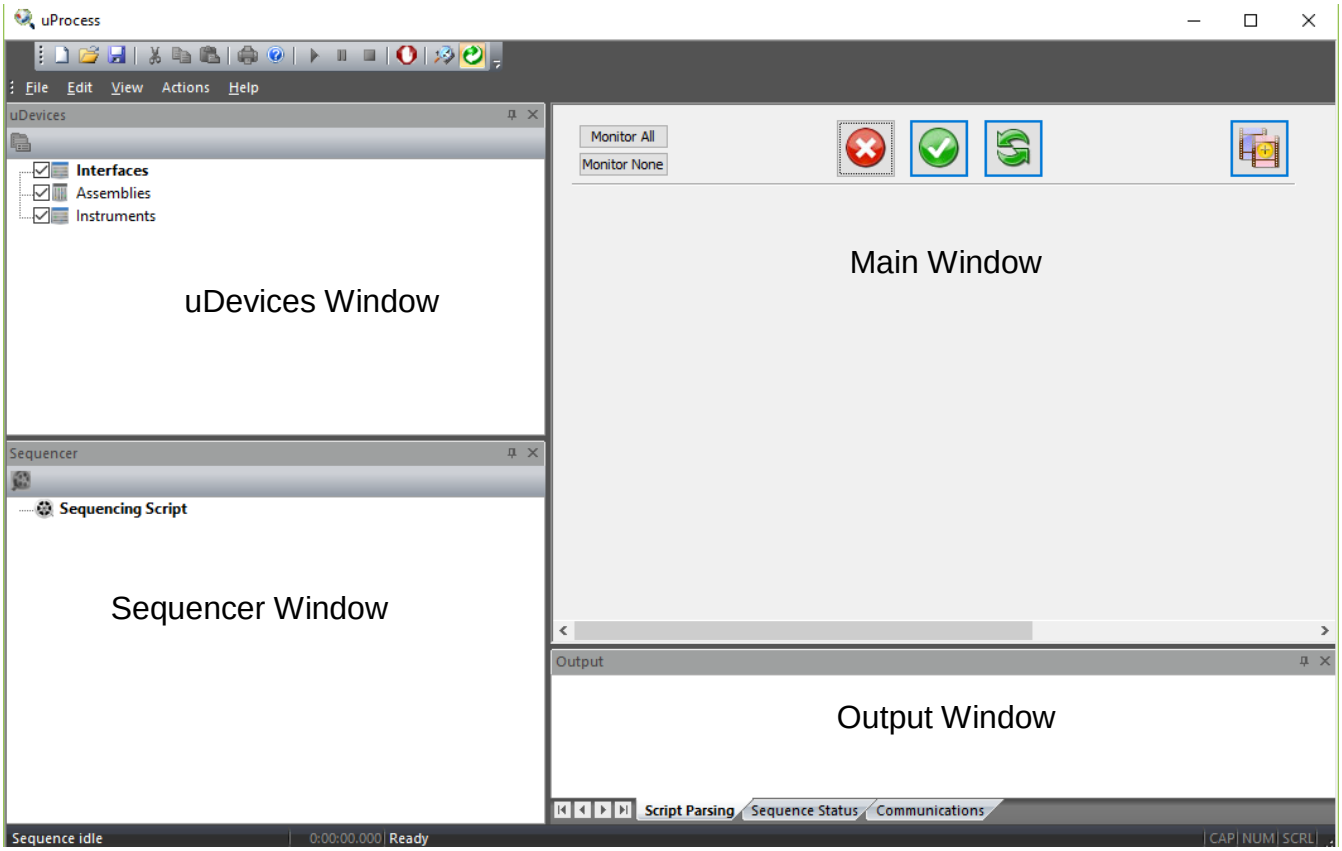


Figure 3. uProcess Dashboard.

The uProcess Dashboard consists of 5 windows, described below. You can customize the placement of the windows on the dashboard and a number of other features about the interface. See ***Customizing the Dashboard*** for more information.

uDevices Window

The uDevices Window is used to view, add, and configure individual devices and EIBs (***Figure 4***). It shares the left side of the Dashboard with the Sequencer Window. Click the tabs at the bottom of this area to switch between the two windows.

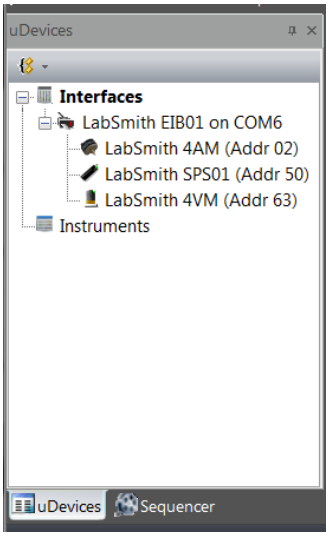


Figure 4. uDevices Window.

Sequencer Window

The Sequencer Window is used to view a sequence program and variable values (***Figure 5***).

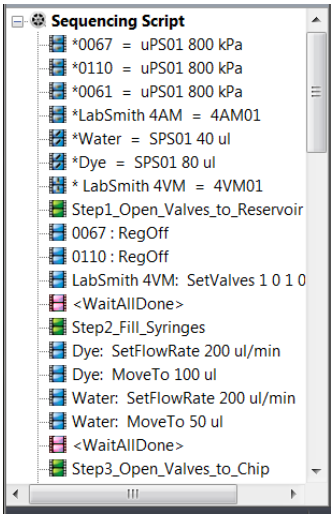


Figure 5. Sequencer Window.

Output Window

The Output Window is used to monitor device communications or to debug and monitor a sequence program (**Figure 6**). By default this window appears at the bottom of the dashboard.

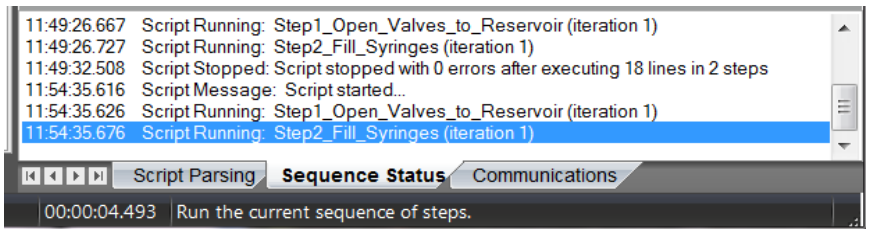


Figure 6. Output Window Showing Script Parsing Tab.

The Output Window contains three tabs:

Script Parsing displays any errors in the current automated process.

Sequence Status indicates which process is running and notes any errors incurred.

Communications shows the status and history of communication between the Electronic Interface Board and the computer.

Main Window

The Main Window is used to control and monitor devices, as well as to construct, edit, and monitor sequence programs. Its contents change depending upon what is selected in the ***uDevices Window*** or ***Sequencer Window***. The Main Window cannot be undocked or hidden.

Communicating with uDevices

Once the uDevice hardware and uProcess software have been installed you are ready to communicate with the devices.

Launch the uProcess software.

Right-click on Interfaces in the uDevices Window and choose New Interface.

Select the computer port to which the EIB is connected, and click **OK.** If multiple EIBs are installed, repeat this step for each EIB.

Your uProcess dashboard should now look similar to ***Figure 7***.

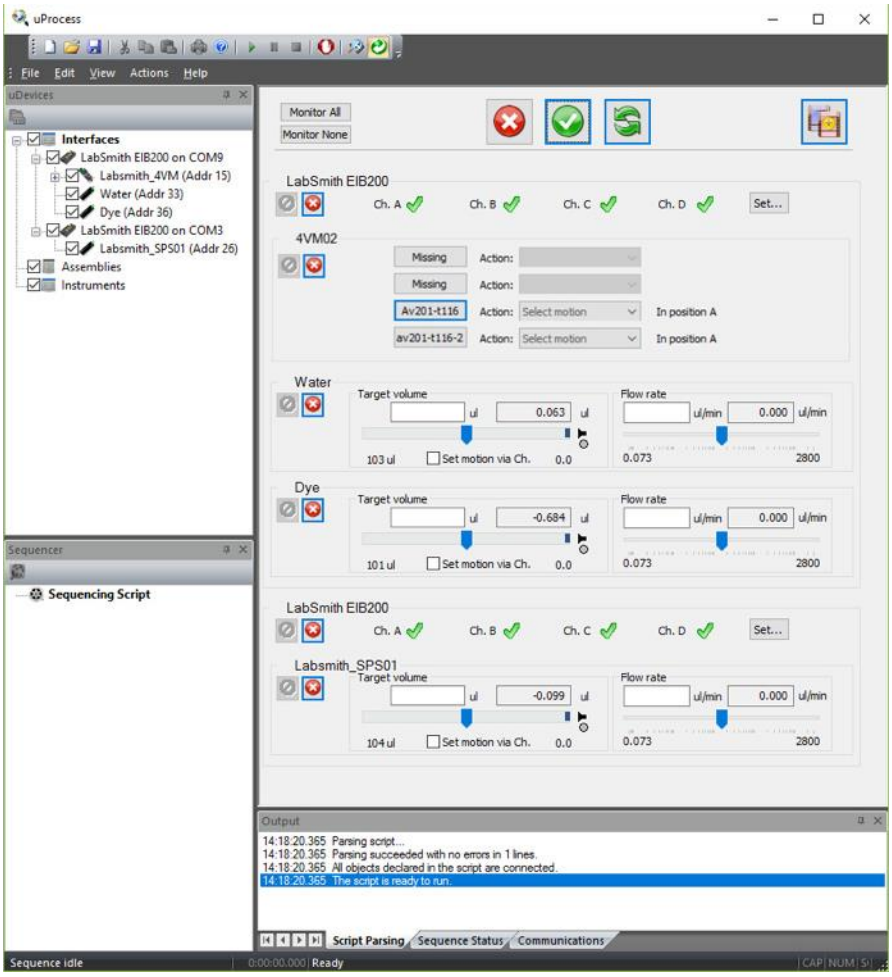


Figure 7. uProcess Dashboard.

The ***uDevices Window*** contains a tree structure listing each EIB and the devices installed under each. In ***Figure 7*** two EIBs are connected. Two SPS01 syringe pumps and a 4VM are connected to the EIB on COM9. A SPS01 syringe pump is connected to the EIB on COM3.

Note: Breadboards will not appear as device in the uDevices tab since there are no functions to control on the breadboard itself.

With the uDevice window selected, the Main Window will show one of two views: the **Control Overview Window** or the **Individual Device Control Panel**.

The **Control Overview Window** allows limited control of all devices. It is also the view that will be used to program sequences using the programming wizard (see ***Create a Script Using the Wizard***).

By clicking on one of the individual device names in the uDevices Window, the **Individual Device Control Panel** opens. ***Figure 8*** shows the control panel for a syringe pump, ***Figure 9*** shows controls for a 4VM01 valve manifold. ***Figure 9*** shows controls for a 4AM01 analog manifold. The **Individual Device Control Panel** provides a wider range of control and setting options for an individual device than is available in the **Control Overview Window.**

Note: Instructions for using the controls in each of the control panels are given in Chapter 4.

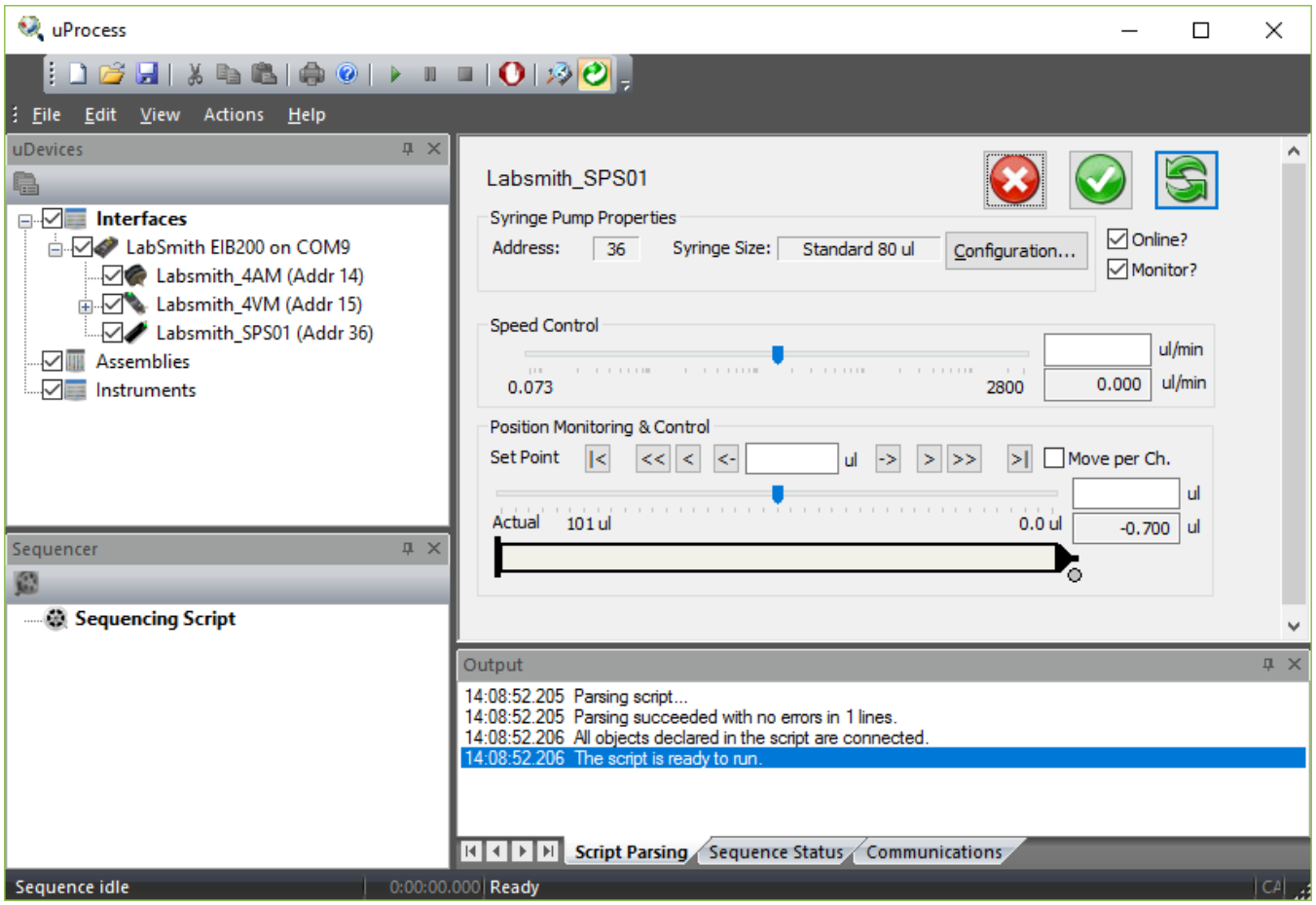


Figure 8. Control Panel for Syringe Pump.

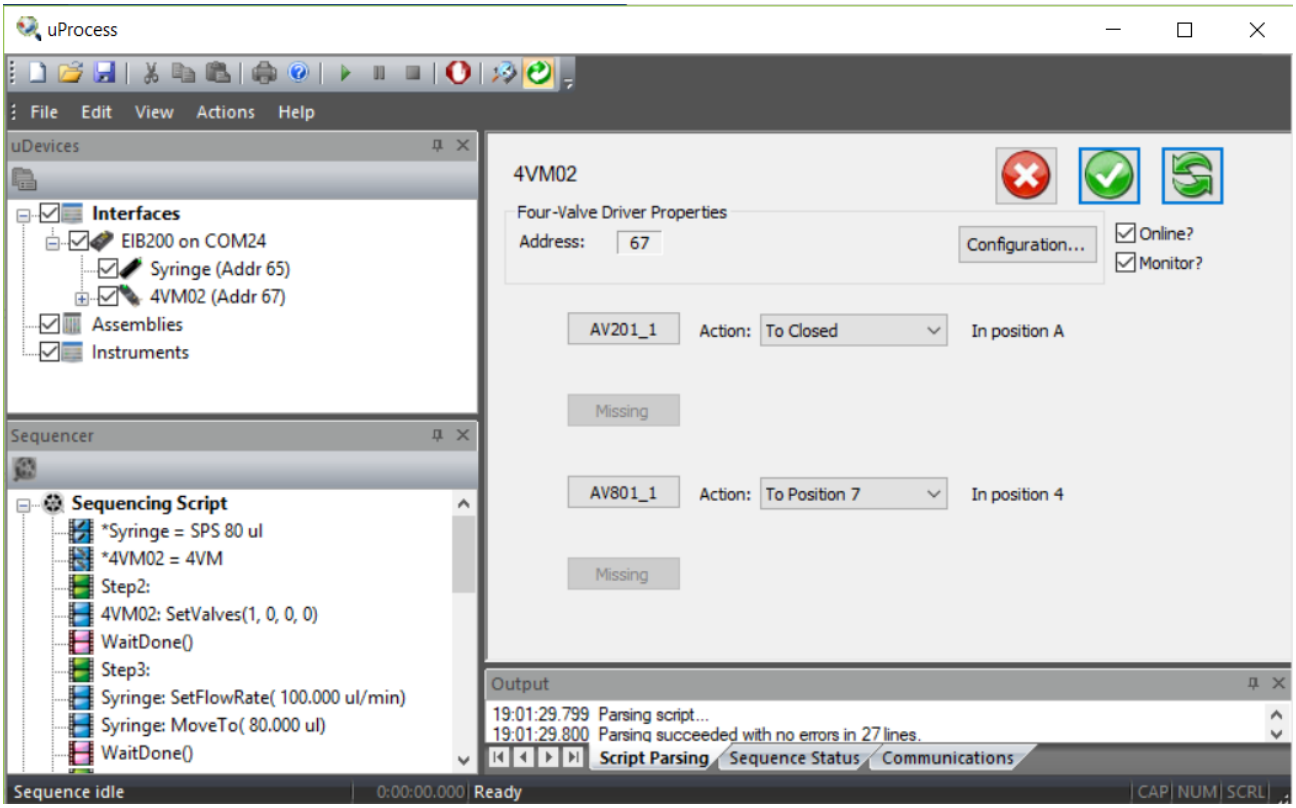


Figure 9. Control Panel for Valve Manifold. Valves installed in position 1 and 3.

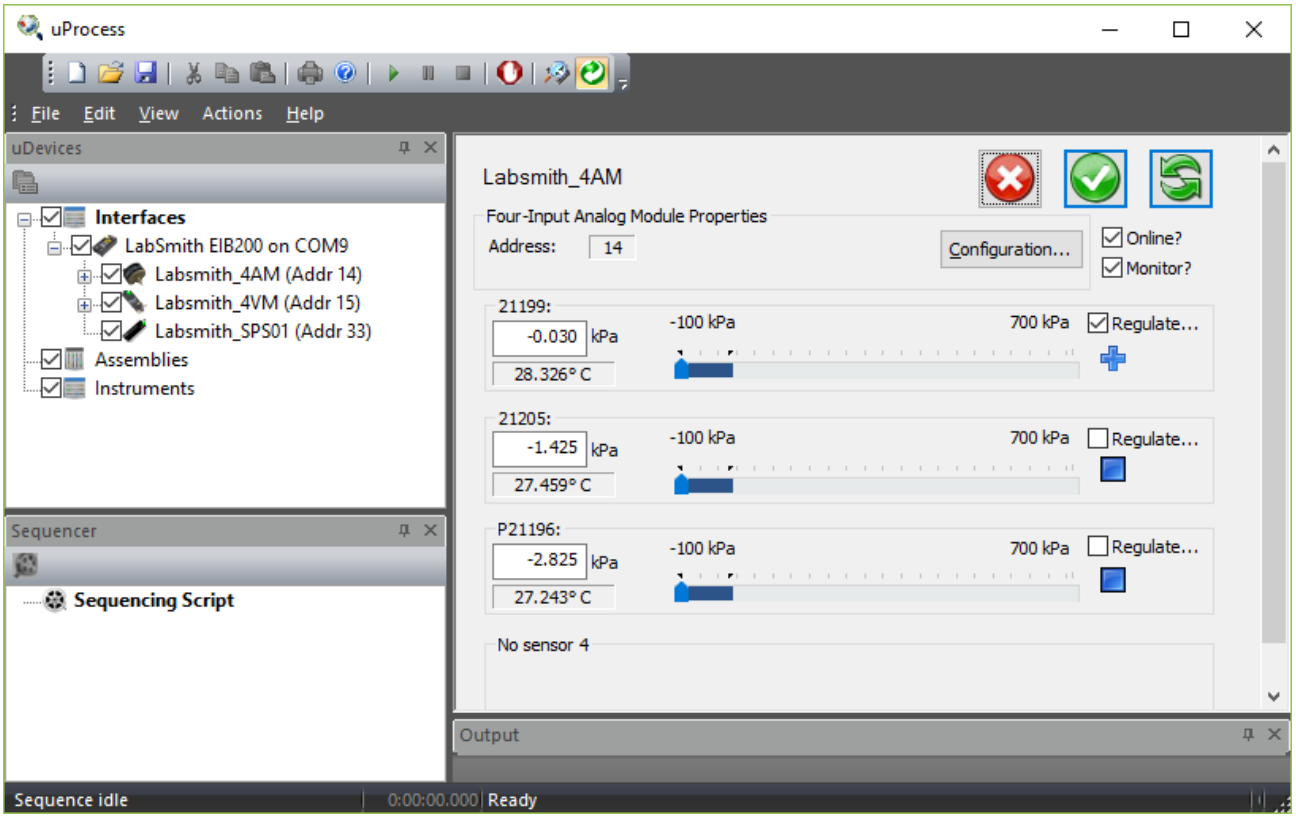


Figure 10 Control Panel for Analog Manifold. uPS pressure sensors installed in positions 1, 2, and 3.

Online and Offline

The communication status of each device is shown in the ***uDevices Window***. To monitor or control a device, the device, its EIB, and the interface must all be online.

As shown in ***Figure 11***, the icon to the left of each device will indicate whether it is Offline (not in communication with the computer) or Online (in communication). In this example the EIB on COM11 is offline, and the EIB on COM1 is online.

To change the status of a device or EIB, right-click the device in the ***uDevices Window*** and check/uncheck the Online setting.

To toggle all connected devices to online or offline, click the **Online** toolbar button .

Note: An EIB will go offline if there is a communication error between the COM port and the EIB. Two red lights on the EIB also indicate a communication problem. This error often indicates a bad cable connection. The **Communications** tab of the Output Window lists the EIB port communication status and history.

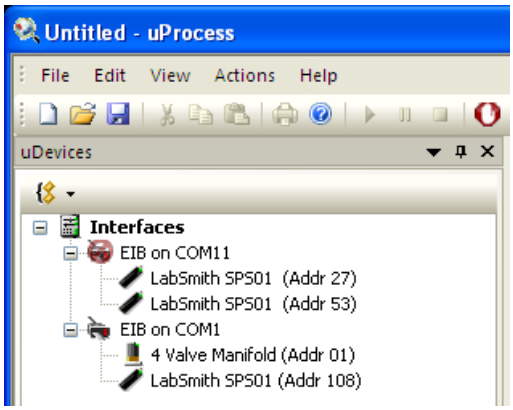


Figure 11 Devices Shown Offline (EIB on Com11) and Online (EIB on COM1).

Device Naming and Numbering

Every uDevice (syringe pump, valve manifold, or analog manifold) is identified by an address and name. The uPS pressure sensors also are individually named

Note: Clicking on the device name in the ***uDevices Window*** will cause the indicator light on the device to flash red/green. This feature aids in quickly identifying each device.

Device Address

Each device attached to an EIB must be assigned a unique address between 0 and 110. If you purchased several devices at the same time from LabSmith they generally will be preconfigured with unique addresses.

Device addresses are shown next to each device in the uDevices Window and also in the Address field of the **Individual Device Control Panels.**

Note: If multiple devices on the same EIB have identical addresses they may not appear in the ***uDevices Window*** list, or they may appear as one device. Operating in this condition can cause unpredictable results.

If two devices with the same address are installed, one will need to be removed before the address can be changed.

Note: The address assigned to each device is independent of its physical position on a breadboard. Devices will be displayed in numerical order on the control panel for each EIB based on the device address, not the position on the breadboard.

To change a device address:

Select the device in the uDevices Window.

Choose Configuration.

Select a **Device Address** from the dropdown menu and click **OK**.

You can also right-click on the device icon in the *uDevices Window* and choose Change Address.

Device Name

Your uDevices will arrive with a default name such as SPS01, 4VM02, or 4AM01.

You can rename the devices. The name may have a maximum of 16 characters and may not include ';', '<', '>', '*', '=', spaces or tabs. Names are case-sensitive.

If you will be using automated processes, each installed device name must be unique.

To change a device name:

Select the device in the uDevices Window.

Choose Configuration.

Type the new device name in the **Name** field.

You can also right-click on the device icon in the *uDevices Window* and choose Rename.

NOTE: Each temperature or pressure sensor arrives with the last five numbers of the serial number as the sensor name. The sensor name can only be changed within the 4AM configuration tab (Please see the Analog Manifold Control section of Chapter 4 for information on changing the sensor name).

Adding or Removing uDevices

To add or remove a device:

- Click the **Online** toolbar button to toggle all interfaces offline.
- Physically add or remove the uDevice(s) to or from the breadboard.
- Click the **Online** toolbar button to toggle all interfaces back online.

Click the **Rescan** toolbar button to check for new devices. Any added devices should now appear in the ***uDevices Window***, while any removed devices will be gone.

Note: To rescan an individual EIB right-click its icon in the *uDevices Window*, then choose Rescan Devices.

Saving Data into a Log File

Time-resolved data can be saved into a data file (format .csv) for later viewing and plotting. Choose File>Log Status/Measurements... to create a Log File. When log file data is being collected there will

be a check mark next to the “Log/Status Measurements...” item in the manu. The file will continue to collect data until you uncheck the box, close the software, or change any of the uDevices (add or remove devices).

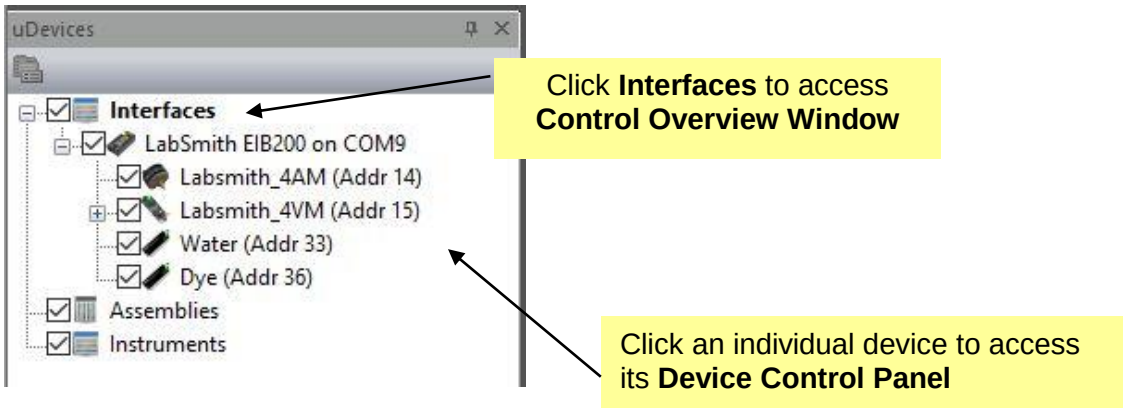
External Instruments

The uProcess software supports several third-party instruments. These instruments are connected to the computer via USB or RS-232 connection and, depending on the instrument, can be monitored and/or controlled. Contact LabSmith support (support@labsmith.com) for information and instructions on supported instruments.

CHAPTER 4. REAL TIME DEVICE CONTROL

uProcess lets you control connected devices manually or as part of an automated process. uDevice control can be accessed two ways:

- through the **Control Overview Window** (accessed by clicking on the Interfaces heading in the uDevices window)
- or through an individual **Device Control Panel**, which can be accessed by clicking on the device name in the uDevices window



The **Control Overview Window** gives you a convenient way to view and set the controls for multiple devices, which is particularly useful when setting up the steps of an automated process, or when controlling multiple devices via pressure regulation. Figure 12 shows a typical **Control Overview Window**. The individual **Device Control Panels** can be selected for additional controls for the particular devices.

Individual uDevices can be controlled via the Device Control Overview Window (shown in Figure 12) or via its individual Device Control Panel. The controls for each type of uDevice are described below.

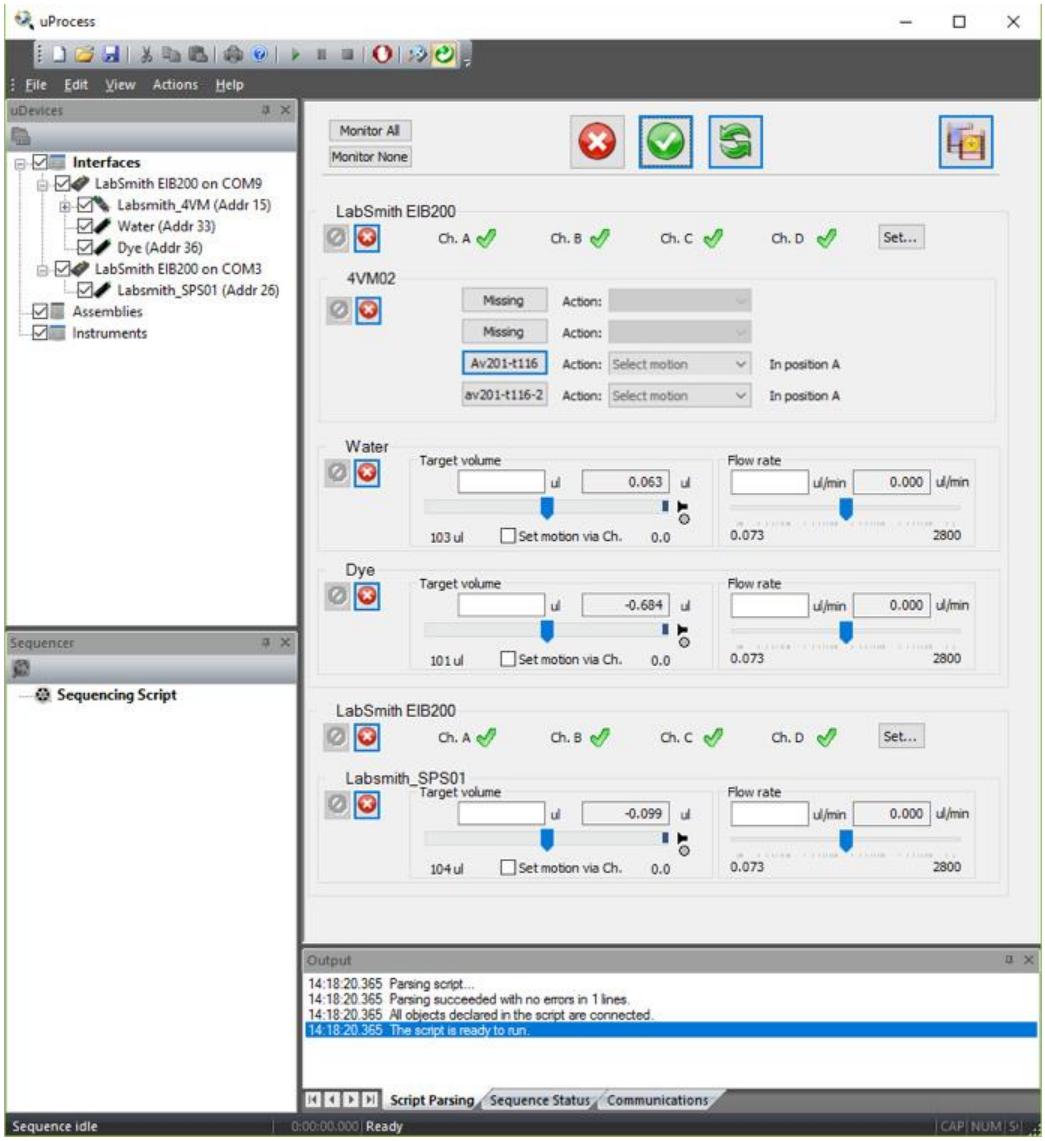


Figure 12 Example of Control Overview Window

4AM Analog Manifold Control

The analog manifold controls are used to monitor and control the uPS pressure sensors and uTS temperature sensors. Typical Control Overview Window and Analog Manifold Control Panel views are shown in Figure 13 and Figure 14, respectively.

Figure 13 Typical Control Overview Window view of analog manifold controls.

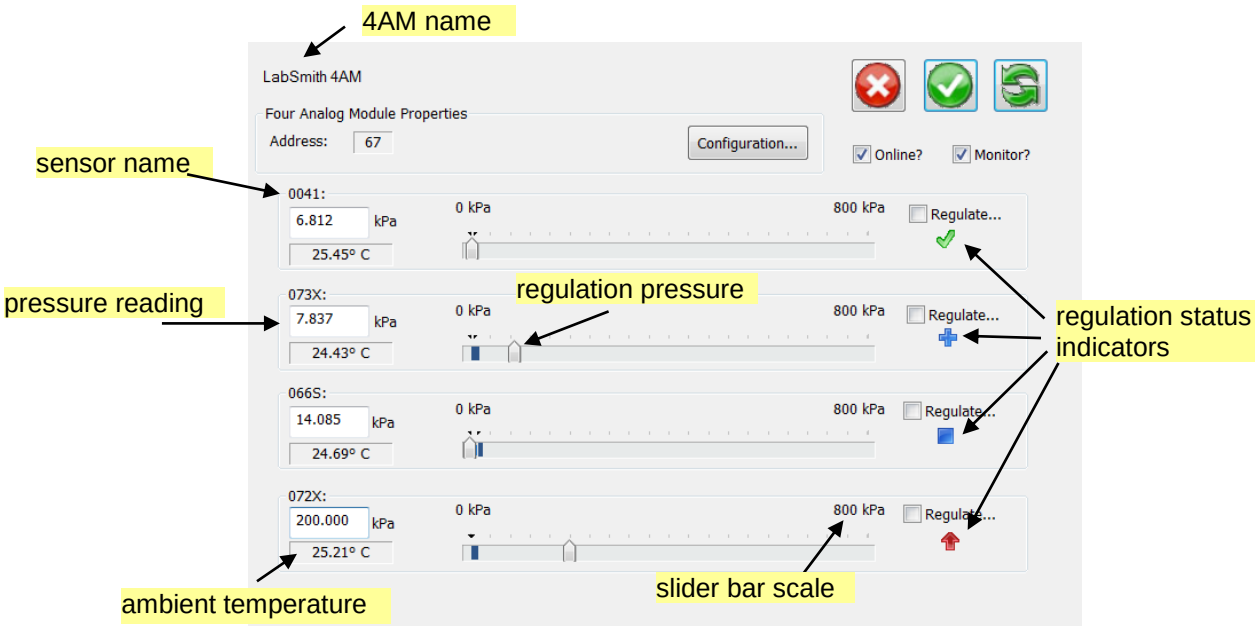


Figure 14 Typical Analog Manifold Control Panel view.

The following commands and information are available for each connected uPS pressure sensor or uTS temperature sensor:

- 4AM01 Name — Set name for the 4AM01
- Sensor Name — Set name for the pressure/temperature sensor (see '**4AM and Sensor Configuration Settings'** for instructions on changing sensor name)
- Current Reading — Current sensor pressure/temperature reading
- Regulation Target Value — Set the regulation pressure/temperature by moving the slider bar, or by typing the desired pressure in the pressure reading box
- Ambient Temperature* — This is the pressure sensor ambient temperature (not the fluid temperature).
- Scale* — Indicates pressure/temperature scale for slider bar (see '**4AM and Sensor Configuration Settings'** for instructions on changing scale)

* ambient temperature and scale are not displayed in the **Control Overview Window**

Pressure and Temperature Regulation

Regulation Status Indicators show the current regulation status (regulating up, regulating down, regulation satisfied, unregulated). See **Channel Regulation** below for more information.

Clicking on "Reg.." or "Regulate..." brings up the Regulation Settings Window, as shown in Figure 15. From this window, you can set the regulation target reading and choose a hardware channel to output the regulation status.

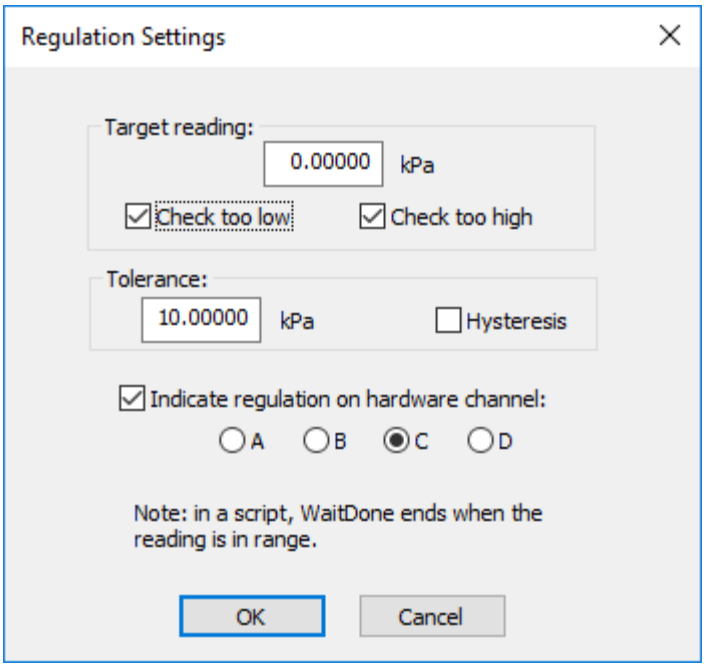


Figure 15 Regulation Setting Window

The following options are available for the Regulation Settings Window:

Target reading: input target value in specified units

- Check too low: the sensor will regulate up to the input value
- Check too high: the sensor will regulate down to the input value, when the actual pressure is equal to or below the target reading the regulation will be complete.

Tolerance: input +/- tolerance for target reading.

- the sensor will regulate until the pressure/temperature value is within the target +/- tolerance

- **Hysteresis**: checking this option causes the pressure or temperature to regulate to the extreme of the target +/- tolerance, and then the regulation stops until the pressure or temperature drifts to the opposite tolerance level

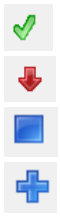
  Note: the tolerance and hysteresis options are only activated if both check too low *and* check too high are selected.

  Note: The Hysteresis option is typically not recommended when controlling temperature.

Channel Regulation: Check the box and select the channel to send the regulation status to an EIB hardware channel. Channel regulation can be used to couple the pressure sensor to a SPS01 syringe pump or automated valve to actuate the syringe or valve based on the pressure regulation status.

If a hardware channel is selected, the channel status will be shown on the EIB control window and the analog sensor panel. **Regulation** Status Indicators include:

Regulating Up



Regulation satisfied

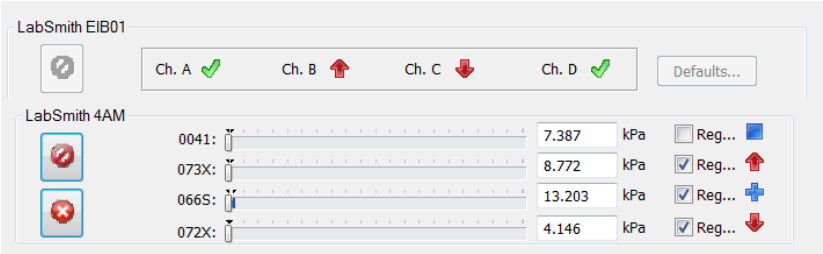
Regulating down

Unregulated

Regulation pending, but not set

For the example shown below:

- uPS 0041 is unregulated
- uPS 073X is regulating up and set to hardware channel B
- uPS 066S is pending, but not set so the regulation type is unknown (the user would press ‘apply settings now’ to set the programming)
- uPS 072X is regulating down and is set to hardware channel C



4AM and Sensor Configuration Settings

Several configuration settings are available to optimize performance of the 4AM01 with pressure and temperature sensors. Click on the ‘Configuration…’ button of the 4AM01 device window to open the *Configure LabSmith_4AM* window shown in Figure 16.

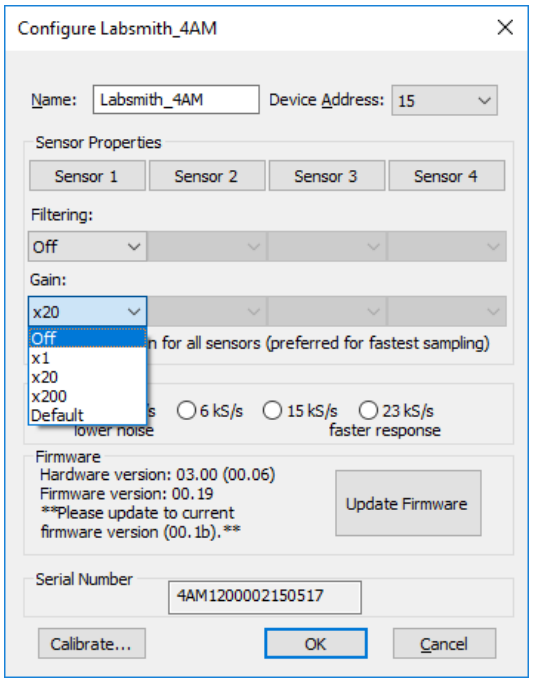


Figure 16 4AM01 Configuration Settings

The gain setting is used to optimize the gain for device and scale range of interest. For the uPS pressure sensors and uTS temperature sensors, the gain readings should be applied as follows:

- Off: does not sample sensor
- x1: not applicable for uPS pressure sensors
- x20: for full-scale pressure readings (default for uPS pressure sensors)
- x200: default rate from uTS temperature sensors; for applications where the pressures will be up to a max of ~10% of full scale

Therefore, for a uPS01-800 pressure sensor, if the measured pressures are expected to be less than 80 kPa, use the x200 gain setting for the most accurate results.

The default sampling rate is 1.8 kS/s. This rate has the lowest noise and will be suitable for most applications.

Zero Offset

The zero reading on the uPS pressure sensors will drift slightly (typically less than ~+/-2 kPa) over time, especially if there are large swings in the sensor temperature. We recommend that you occasionally re-zero the sensors to eliminate the zero offset.

Procedure:

1. Power up the sensors and 4AM and wait for about 5 minutes for the 4AM to warm up.
2. Click on each sensor to bring up its Sensor Properties window (Figure 17)
3. Click the 'Zero Reading' button to zero the pressure sensor.

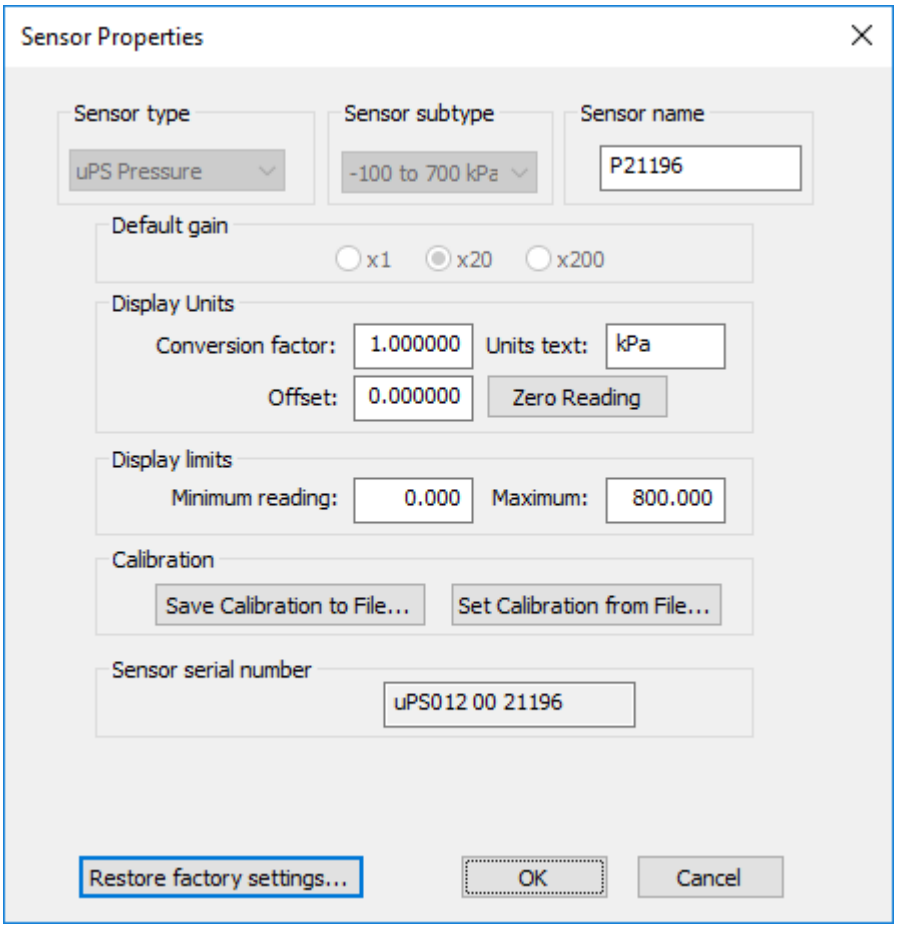


Figure 17 Sensor Properties Window

Note: adjusting the zero offset from this Sensor Properties window will preserve the original factory calibration. The sensor offset can also be adjusted from the main menu at Action>Sensor Calibration>Tare/Set Current Reading. Adjusting the offset from this menu will change the calibration file data.

The *Sensor Properties Window* provides the following additional controls

Sensor Name: The default name is the last 5 numbers of the Sensor serial number

Display Units: The default units are kPa/°C, a conversion factor can be input to display the pressure in units other than kPa/°C

Display limits: these are the limits displayed in the *Control Overview Window* and the *Analog Manifold Control Panel*

The pressure and temperature sensors are shipped with a factory calibration, please contact LabSmith before changing the calibration or factory settings.

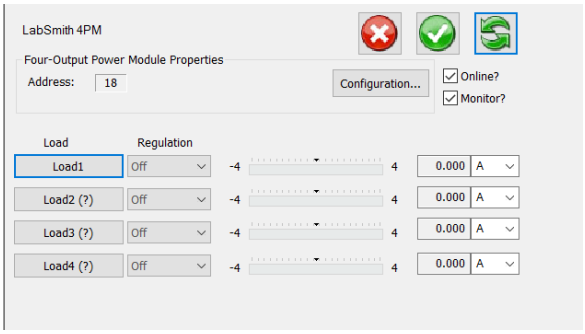
4PM Power Manifold Control

The power manifold controls are used to administer precision heating and cooling using uTE thermo-electric modules.

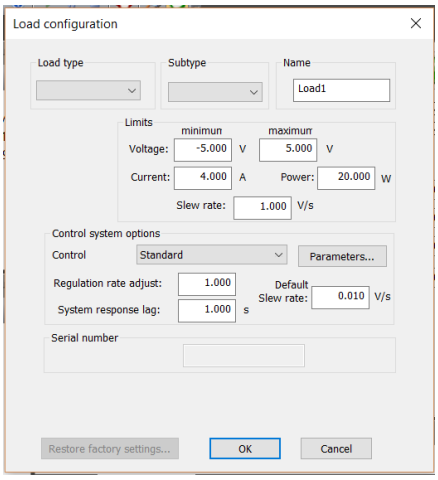
Set the uTE01 Load Type

When using a uTE01 load, the load configuration must be set in uProcess.

1) To set the load configuration, click on the 4PM icon in the Interface tree to bring up the **4PM Module window.**



2) Click on the *Load* button for the channel that the uTE01 is connected (in this case channel 1, or Load1).



3) Using the drop-down menus, set the Load type to *uTE Peltier*, and then select the appropriate Subtype from the drop-down window.

4) Optional: Change the Name for more desciption. For example, since a 13-2020 uTE Peltier is used, the Name is changed to uTE2020.

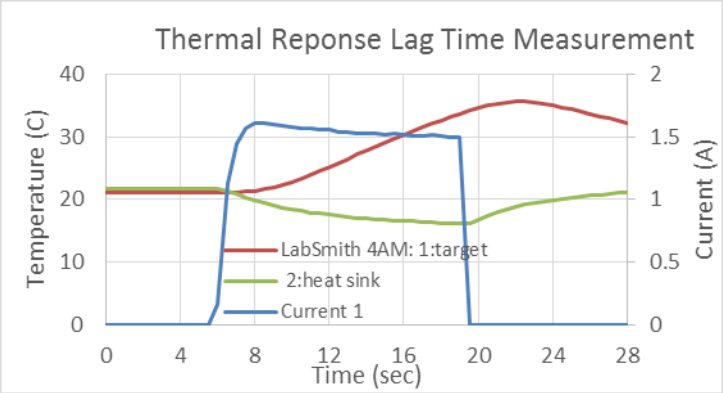
Note: The Load settings are saved in the 4PM memory. If a different type of load is used in the 4PM channel, or if the load is moved to a different channel, the 4PM Load Configuration must be updated. The Load Configuration sets the channel voltage and power limits; using an incorrect configuration can damage the uTE01 Peltier.

Measure the Thermal Response Lag Time

The response lag time is the time delay between applying power to the **uTE controller** and when the temperature change is observed at the target location. The uTE controllers are shipped with a default thermal response lag time of 0.5 seconds. The actual lag time is application dependent and will vary with the location of the thermal sensor. For the most accurate temperature feedback control, the lag time should be measured and adjusted for each application.

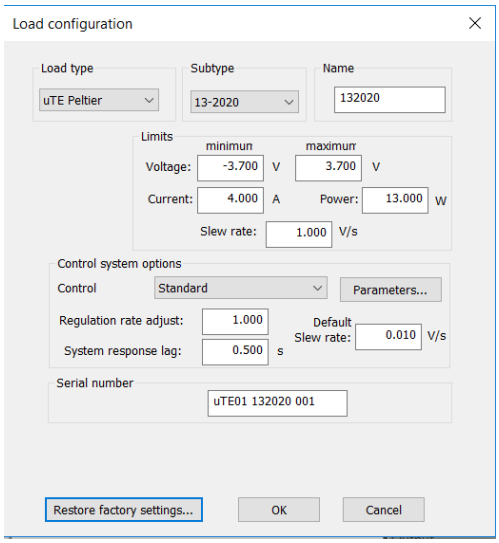
For this example, the uTS temperature sensor (renamed ‘target’) is attached at the target location.

1) To measure the thermal lag time, select File>Log Status/Measurements… to start recording a log file. Select a file name and location when prompted to enter (if not prompted, the log is set as autoname; go to View>Status/Meas Logging Options… to change).
2) Click on the **4PM** in the interface tree to bring up the 4PM uDevice™ window.
3) Select *Power* in the Regulation drop-down menu to set the regulation mode to constant power.
4) Set the power to 2 W and apply settings using the *Apply All Settings Now* button ( ).
5) The target temperature will rise, and the heat sink temperature will fall. Wait until the target temperature has increased approximately 10°C then click *Stop All uDevices* ( ).
6) Go to File>Log Status/Measurements… to uncheck and stop recording the log file.
7) Open the **Log File** in Microsoft Excel or another data processing software, and plot the target temperature and current vs. time. If your uTE is plugged into channel 1 of your 4PM, select *Current 1*. Note the time from when the power is stopped to when the temperature stops rising. This is your thermal response lag time. The figure below shows an example lag time of 3 seconds.



8) Select the **4PM** from the Interface tree to bring up the 4PM control window.

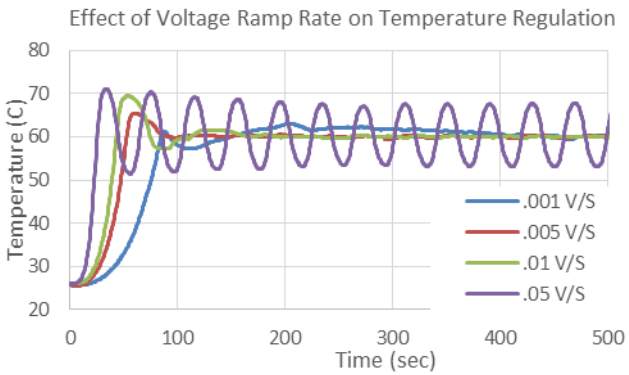
9) Click on the Load Configuration and select the appropriate load (13-2020 in this example).



10) Change the System response lag time to the correct value and click OK.

Selecting a Voltage Ramp Rate for Temperature Regulation

When regulating temperatures, the voltage ramp rate (V/s) can be adjusted to optimize temperature ramp rate, overshoot, and error. The figure below shows various voltage ramp rates for a set temperature of 60 °C.



If precise temperature regulation is required, use a low the voltage ramp rate (~.001 - .005 V/s). If the time to reach the set temperature is crucial, a higher ramp rate should be used (~.01 V/s). If both speed and accuracy are critical, a multi-step regulation routine can be written.

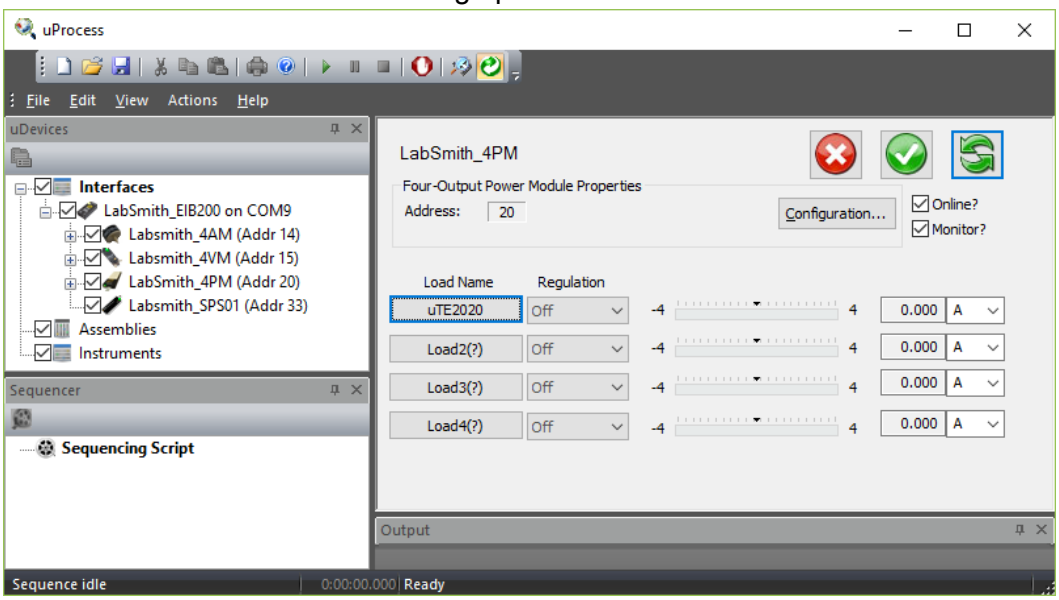
Active vs. Passive Cooling

The uProcess™ thermal products support both active cooling and passive cooling. The default voltage settings for the uTE controllers allow both positive and negative voltage for heating and cooling, respectively. Caution must be used with active cooling (negative voltage), as the back side of the Peltier module can easily be overheated if the steps are not taken to cool the module. If active cooling is to be used, it is best to use a heat sink, fan, and a temperature sensor mounted between the Peltier and heat sink.

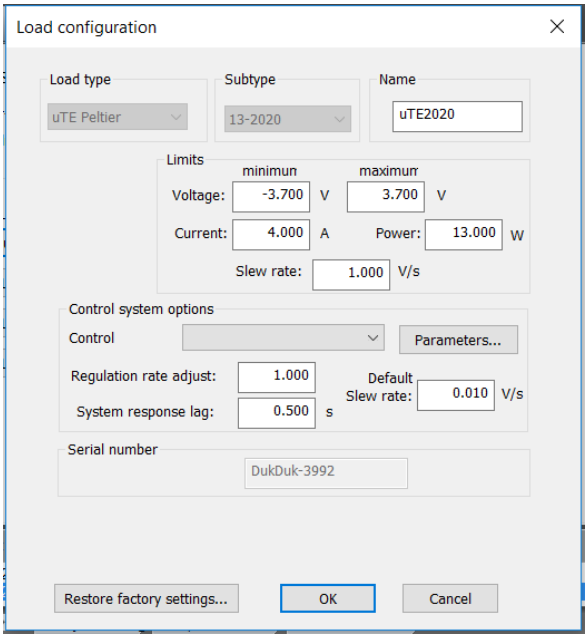
Passive cooling is sufficient for many applications where the application temperatures are above ambient. With passive cooling, the uTE controller minimum voltage is set to 0, and the uTE voltage is reduced or turned off. This eliminates the possibility of overheating the back side of the uTE Peltier.

Changing the uTE controller voltage settings:

1) Click on the **4PM** in the Interface tree to bring up the 4PM uDevice™ Window.



2) Select the load to change (in this case uTE2020).



3) Change the minimum voltage to 0 and click OK.

SPS Syringe Pump Control

Syringe pump control of the set volume and flow rate, and the pressure regulation settings can be manipulated via the **Device Control Overview Window**, as shown in Figure 18. The **Syringe Pump uDevice Control Panel** (Figure 19) provides additional position control options and Configuration settings.

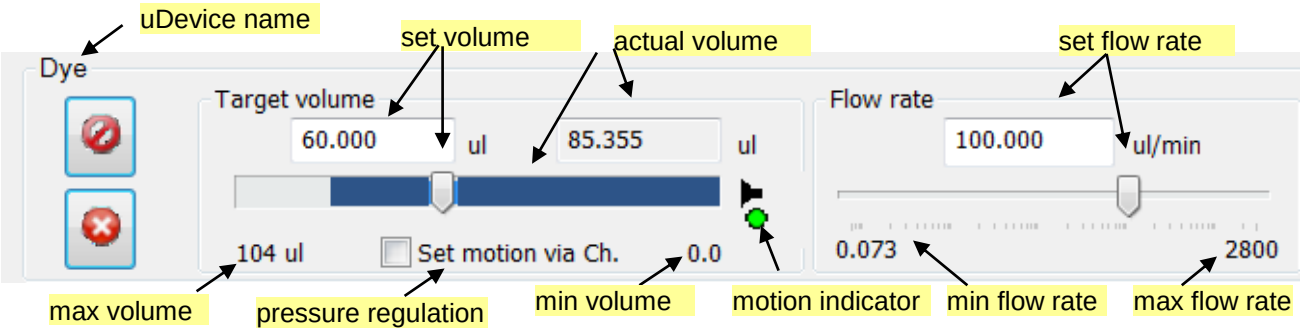


Figure 18 Syringe pump control options in the Device Control Overview Window

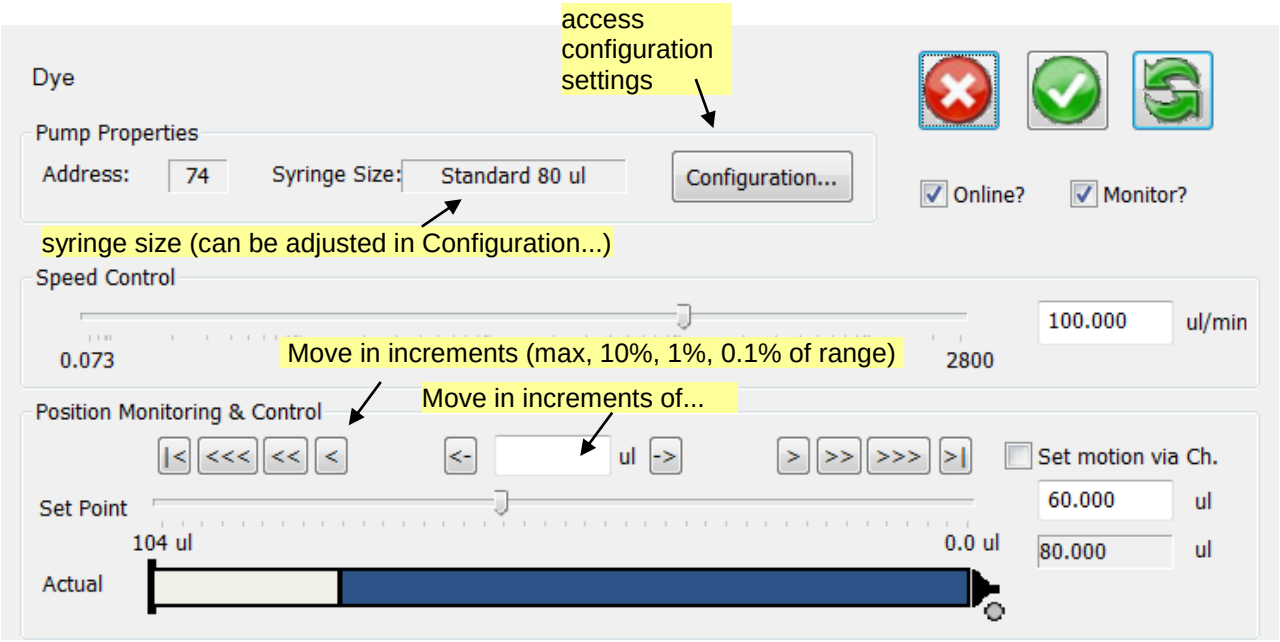


Figure 19 Additional syringe pump control options available in the Syringe Pump uDevice Control Panel.

Target Volume is set by dragging the volume slider, or entering the desired volume in the Target volume box.

Flow Rate is set by dragging the flow rate slider, or entering the desired rate in the Flow rate box.

Note: If **Update All uDevices Continuously** ( ) is selected, changes in target volume or flow rate are immediately applied. If not selected, the pump settings will not change until **Apply All Settings Now (**) or is selected.

The **Motion Indicator** shows the pump motion

green indicates that the pump is pushing

red indicates that pump is pulling

grey indicates the pump is stationary

a red X next to the syringe tip indicates the pump is stalled

The syringe pump can be used to regulate pressure by clicking on the Set motion via Ch. until the desired Channel setting is selected. The Channel Setting options are:

Move with Ch. X

Syringe pump will “push” when pressure on Ch X is below the desired setting.

Syringe pump will “pull” when pressure on Ch X is above the desired setting.

Move opp. Ch. X

Syringe pump will “pull” when pressure on Ch X is below the desired setting.

Syringe pump will “push” when pressure on Ch X is above the desired setting.

Note: Clicking on the target volume input box or moving the volume slider will disable the pressure regulation.

See Analog Manifold Settings for more information on regulating a uPS pressure sensor via a channel setting.

Optimizing Syringe Pump Performance

The syringe pump is designed for optimal accuracy in the push (infuse) direction. The pump motor design includes an anti-backlash spring, which can cause minor volume inaccuracy in the pull direction. In addition, when changing the syringe direction from pull to push, or vice-versa, you may notice a slight delay before the syringe plunger starts advancing. The volume inaccuracy and duration of the delay will vary depending on the syringe size, set speed, system back pressure, and fluid viscosity.

To improve the timing and volume accuracy of an infusion we recommend pushing a small volume of fluid prior to starting the timed infusion. For example, if you want to pull 10 ul of fluid from a reservoir and then push (infuse) it into a process, we recommend pulling 11 µl of fluid, then push 1ul back into the reservoir before switching your valve to the infuse position.

4VM Valve Manifold Control

Click a valve manifold in the uDevices Window to view its control panel (Figure 20).

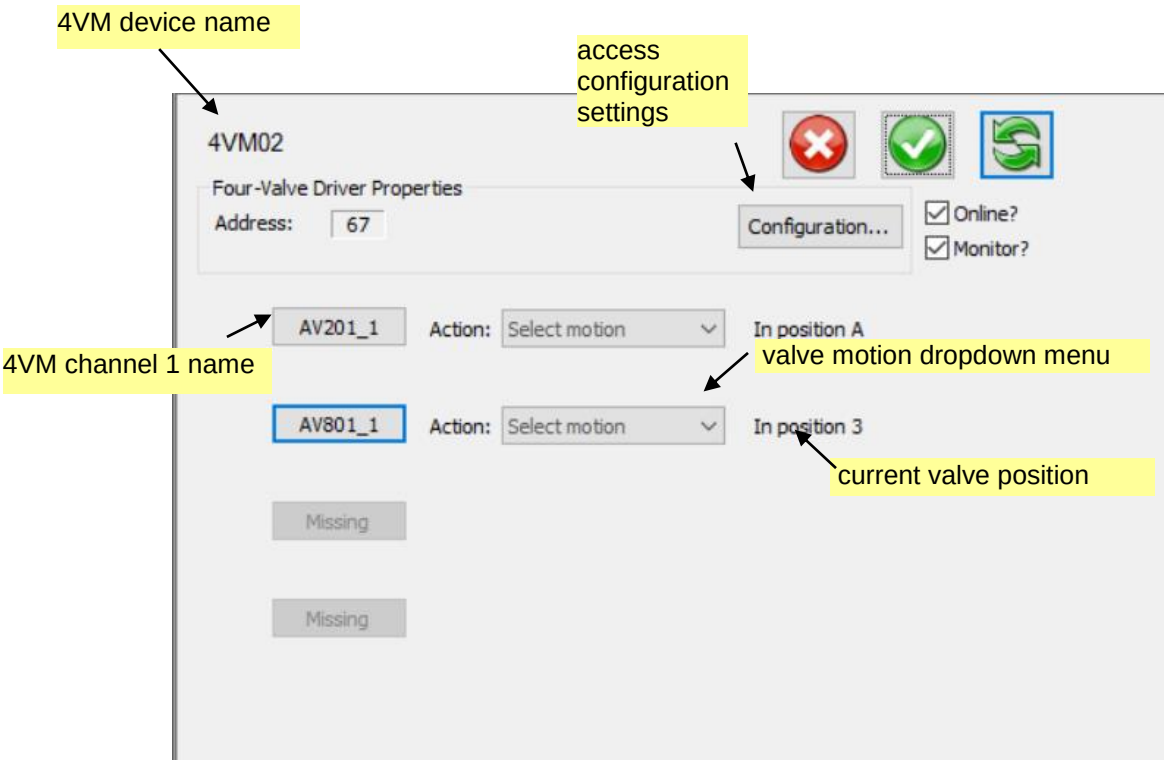
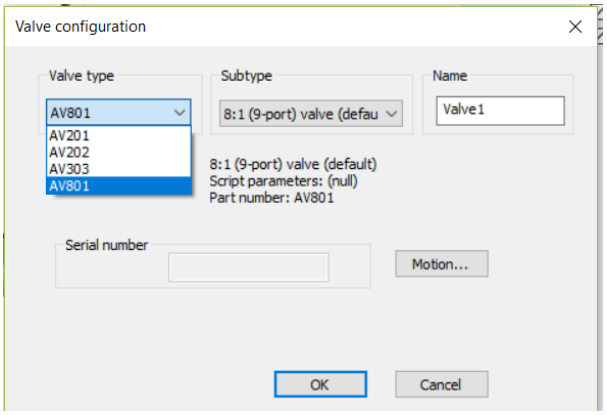


Figure 20. Controls for a 4VM valve manifold, shown with valves connected to Channels 1 and 2.

The valve manifold has four channels. Each of the channels must be configured to the correct valve type before use. Once the valve type has been set for a 4VM channel that type will be saved in the 4VM. If a different valve type is connected to the channel the 4VM configureaition must be updated to reflect the correct valve type.

Click on the channel name to configure the channel.



Valve type: The correct valve must be selected from the dropdown menu for proper valve operation

Subtype: Optional

Name: Optional

Serial number is not currently configured. The *Motion* or *Configuration* buttons will bring up the default valve actuation settings, contact LabSmith before changing these default settings.

Once the 4VM channel has been properly configured, select the *Action* dropdown menu to actuate the valve. Figure 21 shows the dropdown menu options for the AV201, AV202, and AV203 valves.

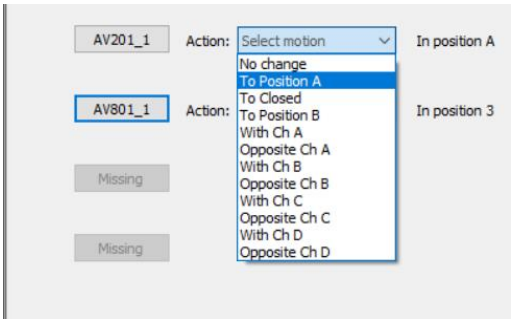


Figure 21. Dropdown menu options for AV201, AV202, or AV303 valves

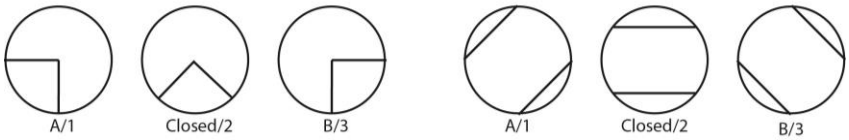
The valve can be actuated to positions A, B, or closed (see Figure 22).

No change The valve state will not change, indicated as position "0" in script.

A The valve will allow flow to the left, indicated as position "1" in script.

Closed The valve will close, indicated as position "2" in script.

B The valve will open flow to the right, indicated as position "3" in script.



AV201 Series (3-port) valve **AV202 Series (4-port) valve**

Figure 22. Shows the valve flow positions, looking down from the front of the valve.

The AV201, AV202, and AV303 valves can also be instructed to set based on the EIB channel regulation status.

Set via Ch. The valve will change states based on the regulation status (see Chapter 5).

If the valve is selected to move "**With**" the channel the default valve positions are:

Regulating up (pressure is too low): valve will move to position "B"

Regulation achieved (pressure is within range): valve will move to position "Closed"

Regulating down (pressure is too high): valve will move to position "A"

If the valve is selected to move "**Opposite**" the channel the default valve positions are:

Regulating up: valve will move to position "A"

Regulation achieved: valve will move to position "Closed"

Regulating down: valve will move to position “B”

Figure 23 shows the dropdown menu options for the AV801 valves.

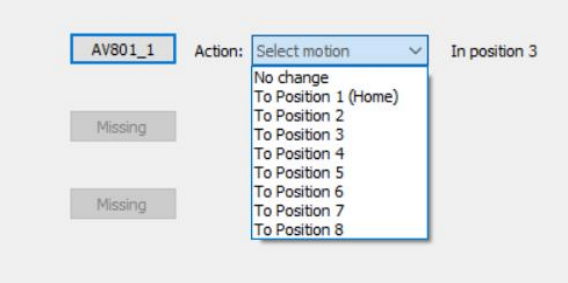


Figure 23. Dropdown menu options for AV801 valve

The indicator lights on the valve will illuminate when it is being actuated. A red light indicates that it is being actuated to the “A” position, while a green light indicates it is being actuated to the “B” position.

If the current valve state is known, there will be a check next to the state and the message below the valve will read “In position”.

If the current valve state is unknown (i.e. if power to the EIB has been cycled), none of the check boxes will be marked and the message below the valve will read “Indeterminate”.

> **NOTE:** If valve cables are switched while the EIB is powered, the valve states could be incorrectly indicated. Remove power to the EIB before changing valves or moving cables.

If **Update Continuously** is selected, the valve will move immediately when you select a new position, and the check mark to the right of that position will be marked. When **Update Continuously** is not selected, choose the new position, then click **Apply Settings Now** to move the valve (the check mark for the new position will be then marked).

CHAPTER 5. AUTOMATED CONTROL

An automated script is a series of steps for controlling one or more automated devices. Each step can change the state of multiple devices. A script can be created using the Process Wizard, and it can be created or edited using the built-in text editor.

Opening Script Files

Each automated script is stored as a file with a .usq extension. To open an existing script click the **Open** toolbar button, then browse for the correct file. Its contents will appear in the **Main Window** (Figure 24).

```
*Reagent_Syringe        =       SPS
*Valve_Manifold1        =       4VM

Fill_Syringe
        Valve_Manifold1 :       SetValves       (0, 0, 0, 1)
        Reagent_Syringe :       SetFlowRate     (50.000 ul/min)
        Reagent_Syringe :       MoveTo          (100 ul)
        WaitDone ()

Deliver_Reagent
        Valve_Manifold1 :       SetValves       (0, 0, 0, 3)
        Reagent_Syringe :       SetFlowRate     (10.000 ul/min)
        Reagent_Syringe :       MoveTo          (0.000 ul)
        WaitDone ()
```

Figure 24. Text of an Automated Process.

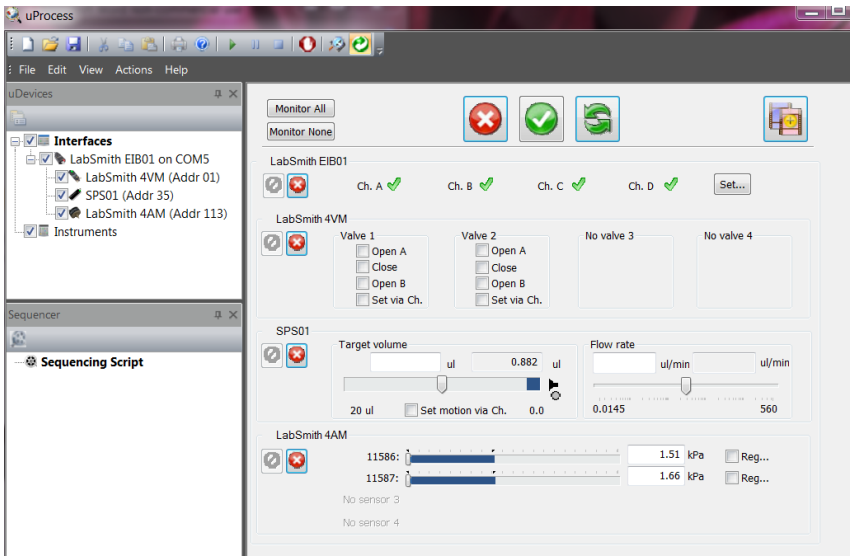
Create a Script Using the Wizard

The uProcess Wizard provides a simplified interface for quickly generating automated scripts.

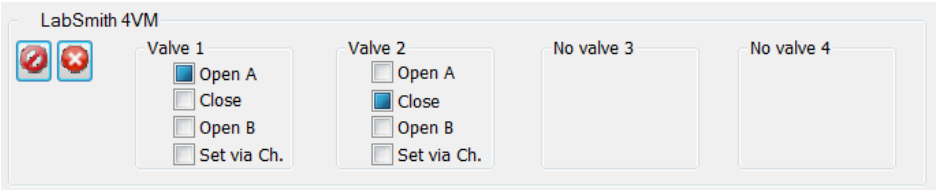
Note: Each installed syringe pump and 4VM01 must have a unique name, regardless of whether it is used in an automated script. See ***Device Name***.

To create a script using the uProcess Wizard:

1. Click **Interfaces** to view the Control Overview Window.



2. Ensure the ‘Update all uDevices Continuously’ button ( ) is NOT selected. If selected, press the stop button ( ) to unselect
3. Select the action for the first step. For example, turn valve 1 to ‘A’ and valve 2 to ‘Closed’.



4. Click the **Save as New Sequence Step** button ( ). The **Creating a New Sequence Step** dialog box will open (Figure 25).

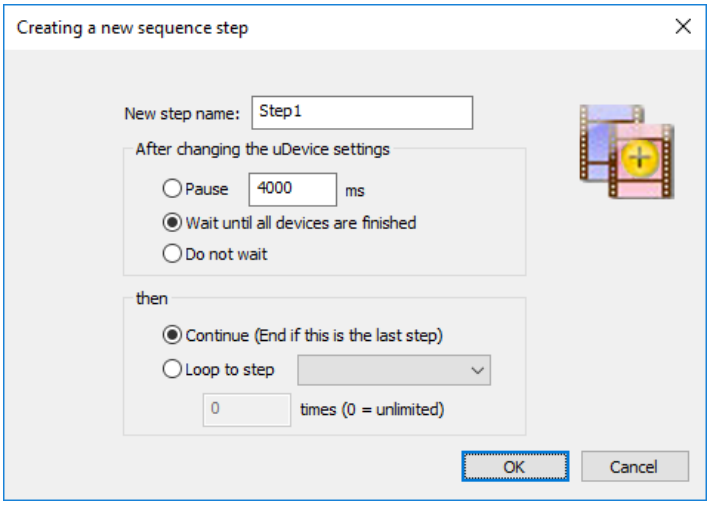


Figure 25. Creating a New Process Step Dialog Box.

5. If desired, enter a descriptive **Step Name**. Step names may have a maximum of 16 characters and may not include blanks or ‘;’, ‘<’, ‘>’, ‘*’ or ‘=’.
6. Select how the script should respond **after changing the uDevice settings**:

   Select **Pause**, then enter the duration to pause before continuing with the process. The pause duration begins as soon as the step starts.

   If you select **Wait until all devices are finished**, uProcess will wait until all devices have completed their functions before continuing on with the process.

   Select **Do Not Wait** to immediately continue on to the next step.
7. Select how the script should continue after this step:

   Select **Continue** to continue to the next step in the process. If a new step is not added the process will end.

   Select **Loop to step** to jump to another process step. Select the target step from the drop-down list, then enter the number of times the step should loop (if 0 is entered loop will be infinate).
8. Click OK. The step will appear in the Sequencer Window, and its text will appear in the Main Window.

Save a Script File

Click the **Save** toolbar button, or choose **File>Save**, to save all changes to the current script file. If the file has not yet been saved you will be prompted to provide a name and file location for it.

Choose **File>Save As** to store the current script under a new file name.

Run, Pause and Quit a Script

When a script is open and error-free, the **Run** toolbar button will be available. Click **Run** to initiate the process. All steps will be performed in order.

While a script is running you can click the **Pause** toolbar button to pause the automated process. Click **Pause** again to continue the script from that point.

Click the **Quit Process** toolbar button to quit running the automated script.

Note: When you Pause or Quit a script, devices that were moving will continue until they reach their target states. Click the **Stop All Devices** toolbar button to stop all device activity.

While an automated script is running, the **Sequence Status** tab of the ***Output Window*** will indicate the current status of the script and will report any errors encountered (Figure 26).

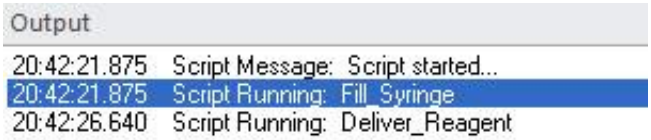


Figure 26. Sequence Progress Tab of the Output Window

.

Automated Script Functions

Beep **Beep()**

Plays an audible sound (the sound associated with the asterisk in System Sounds).

Example: `Beep()`

Note: The **Beep** function cannot be added using the Process Wizard.

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 Process Wizard

Goto **Goto** *StepName*

Creates a loop back to named step.

StepName: defined name of step in script

Example: `Goto FillStep`

HysteresisReg *DeviceName*: **HysteresisReg** (*Pmin units, Pmax units)*

Regulates between two pressures. If pressure starts below regulation range, uProcess will regulate up to *Pmax* and then let the pressure drift down to *Pmin*. If pressure starts above regulation range, uProcess will regulate down to *Pmin* and then let the pressure drift up to *Pmax*.

DeviceName: defined name of uPS pressure sensor (must be defined in script header)
Pmin: minimum pressure in range
Pmax: maximum pressure in range
units: pressure units

Example: `HysteresisReg (40.000 kPa, 60.000 kPa)`

HysteresisReg to Channel

DeviceName: **HysteresisReg** (*Pmin units, Pmax units,* **DC*h****X****)*
SyringeName: **MoveWith (DCh*X*)** *or* *SyringeName:* **MoveOpposite (DCh*X*)**

Similar to **HysteresisReg**, except it sends the output value to a specified channel that can be used to control a SPS01 syringe pump. Requires two lines of code:

DeviceName: defined name of uPS pressure sensor (must be defined in script header)
Pmin: minimum pressure in range
Pmax: maximum pressure in range
units: pressure units

SyringeName: defined name of syringe pump (must be defined in script header)
X: channel *A, B, C, or D.*

MoveWith (DChX): pulls the syringe to decrease the pressure
pushes the syringe to increase the pressure

MoveOpposite (DChX): pushes the syringe to decrease the pressure
pulls the syringe to increase the pressure

Example:
```
HysteresisReg (40.000 kPa, 60.000 kPa, DChA)
SPS01:      MoveWith (DChA)
```

IfDone

DeviceName: **IfDone()**

If regulation is finished, the immediately following step will be executed; otherwise, the immediately following step will be skipped.

DeviceName: defined name of uPS pressure sensor (must be defined in script header)

Example:
```
uPS1: IfDone()
Beep()
```

Note: The **IfDone** function cannot be added using the Process Wizard

IfNotDone

DeviceName: **IfNotDone()**

If regulation is not done, the immediately following step will be executed; otherwise, the immediately following step will be skipped.

DeviceName: defined name of uPS pressure sensor (must be defined in script header)

Example:
```
uPS1: IfNotDone()
Goto Step1
```

Note: The **IfNotDone** function cannot be added using the Process Wizard

Log

log (**on)**
log (off)
log (*optional filename***)**

Opens/closes a log file.

filename: file path and name (see examples)

If omitted, the script will promt you for a file name. Auto file naming can also 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)
```

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 FillStep 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 (must be defined in script header)
vol: desired syringe volume
units: volume units, acceptable units include: ul (default), ml, and nl

Example:
```
Reagent:    MoveTo (20.5 ul)
```

Wait

Wait (*duration units)*

Waits for specified duration until moving on to the next line.

Duration length of wait
units wait duration units, acceptable units include: ms (default), s, and us

Example:
```
Wait (1 s)
```

SetChannels

*EIB200_Name***: SetChannels** (*x, x, x, x)*

Sets EIB200 Channels.

EIB200_Name: defined name of EIB200 (must be defined in script header)
x, x, x, x position of channels A, B, C, and, D, respectively, where
- x = 0: no change
- x = 1: regulate up
- x = 2: no regulation
- x = 3: regulate down

Example:
```
EIB200:     SetChannels (1, 3, 0, 2)
```

SetCurrent *4PM_Name*: **SetCurrent** (*output,* value *units)*

Sets current of the output port.

4PM_Name	defined name of 4PM (must be defined in script header)
output	Output supply channel (1-4). Constants 'Out1', 'Out2', Out3', or 'Out4' can be used.
units	power units, acceptable units include: A (default)

Example:
```
4PM:  SetCurrent (1, 2 A)     ; both are the same command
4PM:  SetCurrent (Out1, 2 A)
```

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, non-zero number)
units:	flow rate units, acceptable units include: ul/min (default), ul/s, nl/min, and nl/s

Example:
```
Reagent:    SetFlowRate (50 ul/s)
```

SetOff *4PM_Name*: **SetOff** (*output)*

Unsets the power, voltage or current.

4PM_Name:	defined name of 4PM (must be defined in script header)
output:	Output supply channel (1-4). Constants 'Out1', 'Out2', Out3', or 'Out4' can be used.

Example:
```
4PM:  SetOff (1)        ; both are the same command
4PM:  SetOff (Out1)
```

SetPerChannel *4PM_Name*: **SetPerChannel** (*output, DChX,* value *units)*

Sets power of the output port.

4PM_Name:	defined name of 4PM (must be defined in script header)
output:	output supply channel (1-4). Constants 'Out1', 'Out2', Out3', or 'Out4' can be used.
Units:	power units, acceptable units include: V/s (default)

Example:
```
4PM:  SetPerChannel (1, DchA, 2 V/s)      ; both are the same
4PM:  SetPerChannel (Out1, DchA, 2 V/s); command
```

SetPower *4PM_Name*: **SetPower** (*channel,* value *units)*

Sets power of the output port.

4PM_Name:	defined name of 4PM01 (must be defined in script header)
channel:	4PM channel (1-4). Constants 'Out1', 'Out2', Out3', or 'Out4' can be used.

Units: power units, acceptable units include: W (default)

Example:
```
4PM:  SetPower (1, 2 W) ; both are the same command
4PM:  SetPower (Out1, 2 W)
```

SetSelection *4VM_Name*: **SetSelection** (*channel,* value *position)*

Sets AV801 valve position.

4VM_Name: defined name of 4VM (must be defined in script header)
output: 4VM channel (1 - 4)
position: valve port to open (1 - 8)

Example:
```
4VM01: SetSelection (1, 4); opens valve in 4VM channel 1 to
port 4.
```

Note: Use **SetValves** command to set position of AV201, AV202, and AV303 valves.

Note: Ensure the correct valve type is selected in the 4VM configuration.

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

Sets AV201, AV202, or AV303 valve positions.

4VM_Name: defined name of 4VM (must be defined in script header)
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_A:    SetValves ()
```

Note: Use **SetSelection** command to set the AV801 position.

Note: Ensure the correct valve type is selected in the 4VM configuration.

SetVoltage *4PM_Name*: **SetVoltage** (*output,* value *units)*

Sets voltage of the output port.

4PM_Name: defined name of 4PM (must be defined in script header)
output: Output supply channel (1-4). Constants 'Out1', 'Out2', Out3', or 'Out4' can be used.
units: voltage units, acceptable units include: V (default)

Example:
```
4PM:  SetVoltage (3, 1 V)     ; both are the same command
4PM:  SetVoltage (Out3, 1V)
```

Stop

DeviceName: **Stop()**

Stops device.

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

Example: `Reagent: Stop()`

Note: The **Stop** command is added via the Process Wizard using the button (Stop this uDevice) in the Device Control Overview window.

Quit

Quit

Quits the script, steps following this function will not be run. This function will not stop a device that is already in process, for that use the **Stop** function.

Example: `Quit`

Note: The **Quit** function cannot be added using the Process Wizard.

RegDownTo and RegUpTo

SensorName: **RegDownTo** (value *units)*; regulation is complete when sensor pressure reaches or drops below input value

or

SensorName: **RegUpTo** (value *units)*; regulation is complete when sensor pressure reaches or exceeds input value

SensorName: defined name of sensor (must be defined in script header)
value: pressure (positive or negative number)
units: pressure units, acceptable units: kPa

Example: `P10032: RegDownTo (50 kPa)`

RegDownTo Channel or RegUpTo Channel

Similar to RegDownTo and RegUpTo, except it sends the output value to a specified channel which is used to control one or more SPS01 syringe pump.

Requires two lines of code:

SensorName: **RegDownTo** (value *units, **DCh**X)*
SyringeName: **MoveWith (DCh***X)* *or* *SyringeName:* **MoveOpposite (DCh***X)*

or

SensorName: **RegDownTo** (value *units*, ***DCh**X)*
SyringeName: **MoveWith (DCh***X***)** *or* *SyringeName:* **MoveOpposite (DCh***X***)**

SensorName: defined name of sensor (must be defined in script header)

value: pressure (positive or negative number)
units: pressure units, acceptable units: kPa
SyringeName: defined name of syringe pump (must be defined in script header)
X: channel *A, B, C, or D*

MoveWith (DChX): pulls the syringe to decrease the pressure
pushes the syringe to increase the pressure

MoveOpposite (DChX): pushes the syringe to decrease the pressure
pulls the syringe to increase the pressure

Example:
```
P032:       RegDownTo (50 kPa, DChA)
SPS01:      MoveWith (DChA)
```

Note: Use the **SetFlowRate** command to set the syringe pump flow rate for each syringe, else the last flow rate set for the syringe will be used.

RegOff *SensorName*: **RegOff()**

Turns off the regulation for a pressure sensor.

Example:
```
032:  RegOff()
```

WaitDone **WaitDone()**

Pauses script until all processes are complete.

Example:
```
WaitDone()
```

General Formatting Rules

Devices

Individual devices are defined in the script as follows:

Syringes:

```
*DeviceName =     SPS01 size ul
```

Where *size* is the syringe size (4, 8, 20, 40, or 80)

Valve Manifolds:

```
*DeviceName =     4VM01
```

DeviceName must match name defined in the uDevice configuration (see ***Device Naming and Numbering***).

Comments

Comments can be added to the script using a semicolon (;) or a pound symbol (#). Anything after the character is treated as a comment. Comments using the semicolon will not show up in the Sequencer, but comments with the pound symbol will.

```
Wait (1000 ms) ; to allow time to monitor reaction
; Beep() - this command will not be executed
# Beep() - this command will not be executed
```

Labels

Labels may have a maximum of 16 characters and may not include blanks or ';', '<', '>', '*' or '='. All lables must be followed by a colon (':').

Labels allow you to jump to a certain part of the code and the software records when you pass a label while running.

Variables

Variable names can include characters, numbers, or underscores. Each variable name must begin with a character or underscore. Variables can be defined and redefined anywhere in the script. All defined variables (including ones of each device defined in the script) are displayed in the Sequencer.

```
a = 10
_b5 = 5
Full = 20
```

Variables cannot have the same name as a defined constant (for a list of these names see **Defined Constants**). Use the following calls to use variables of defined devices in the script.

Instrument	Variable call	Type of variable	Description
All Instruments	deviceName.IsDone	True/false	Instrument has completed regulation, moving, or switching
Syringe Pump	deviceName.FlowRate	Int	Last syringe flow rate
	deviceName.IsMoving	True/false	True if syringe is moving
	deviceName.IsStalled	True/false	True if syringe is stalled
	deviceName.MaxFlowRate	Int	Maximum syringe flow rate (2816)
	deviceName.MaxVolume	Int	Furthest fill position of syringe
	deviceName.Volume	Int	Current position of syringe
Pressure Sensor	deviceName.Reading	Int	Pressure reading
	deviceName.Reading2	Int	Temperature reading
Temperature Sensor	deviceName.Reading	Int	Temperature reading 1
	deviceName.Reading2	Int	Temperature reading 2
Valve	deviceName.Pos	1, 2, or 3	Valve Position

```
If (uPS.Reading >=50)
      SPS01:            SetFlowRate (SPS01.MaxFlowRate ul/min)
```

Variables can be manipulated using math functions.

Function	Description	Example
'+'	Adds two numbers or variables	`a = a + 1` `b = c + a`
'-'	Subtracts two numbers or variables	`a = a – 1` `b = c – a`
'*'	Multiplies two numbers or variables	`a = a * 5` `b = c * a`
'/'	Divides two numbers or variables	`a = a / 2` `b = c / a`
'^'	Raises a number or variable to a power	`a = a ^ 2` `b = c ^ a`
'sqrt'	Takes the square root of a number or variable	`a = sqrt(125)` `b = sqrt(a)`
'sin'	Takes the sine of a number or variable	`a = sin(0)` `b = sin(a)`
'cos'	Takes the cosine of a number or variable	`a = cos(pi)` `b = cos(a)`
'tan'	Takes the tangent of a number or variable	`a = tan(0)` `b = tan(a)`
'sinh'	Takes the hyperbolic sine of a number or variable	`a = sinh(0)` `b = sinh(a)`
'cosh'	Takes the hyperbolic cosine of a number or variable	`a = cosh(pi)` `b = cosh(a)`
'tanh'	Take the hyperbolic cosine of a number or variable	`a = tanh(0)` `b = tanh(a)`
'max'	Returns the maximum value of a set of numbers or variables	`a = max(19,34,25,72,4)` `b = max(x, y, z, 23)`
'min'	Returns the minimum value of a set of numbers or variables	`a = min(19,34,25,72,4)` `b = min(x, y, z, 23)`
'abs'	Returns the positive version of a number or variable	`a = abs(-3)` `b = abs(-a)`
'log'	Takes the log base 10 of a number or variable	`a = log (100)` `b = log (a)`

Multiple math functions can be performed in the same line. Each function will follow the order of operations, not the order listed in the line.

Our functionality follows the open-source MIT licensed software package "muParserX" to perform symbolic math. This software parses text into a reverse-polish-notation stack of byte codes that perform mathematical and string operations at near-compiled speeds.

Operators

Operators compare two numbers or variables and return true or false.

Operator	Use	Example
'=='	"is equal to"	`If (a == 5)`
'!='	"is not equal to"	`If (a != 5)`
'>'	"is greater than"	`If (a > 5)`
'<'	"is less than"	`If (a < 5)`
'>='	"is greater than or equal to"	`If (a >= 5)`
'<='	"is less than or equal to"	`If (a <= 5)`

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 the condition is true.
If (*condition*)
{
    Statement ; using brackets you can have more than one statement that runs if the
    Statement ; condition is true.
}
x = (*condition*)? *value1* : *value2* ; if the condition is true, x will be assigned *value1* if false, x ; will be 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
}
```

Timer

Use the scripttime() function to time how long functions take.

Example:

```
start_time = scripttime()
Wait(10 s)
end_time = scripttime()
time_elapsed  = end_time - start_time
```

Defined Constants

These globally defined constants can be used in the script in place of the values they represent. They cannot be reassigned to a different value.

Constant Name	Value	Constant Name	Value
‘pi’	3.14…	‘Out2’	2
‘e’	2.71…	‘Out3’	3
‘DChA’	1	‘Out4’	4
‘DChB’	2	‘In1’	1
‘DChC’	3	‘In2’	2
‘DChD’	4	‘In3’	3
‘On’ or ‘on’	1	‘In4’	4
‘Off’ or ‘off’	0	‘PosA’	1
‘None’	0	‘PosB’	3
‘Out1’	1	‘PosClosed’	2

Editing Automated Script Text

As mentioned earlier, you can create a script using the Process Wizard, or you can create/edit the text directly.

To view and edit the text of a script, click on the **Sequencer** window, then select **Sequencing Script**. The text will appear in the **Main Window**. Here is a sample script:

```
*Syringe    =           SPS01
*Manifold   =           4VM01
Open_Reservoir:
      Manifold:   SetValves (1, 0, 0, 0)
      WaitDone()
Fill_Syringe:
      Syringe:    SetFlowRate (50 ul/min)
      Syringe:    MoveTo (10 ul)
      WaitDone()
Open_Process:
      Manifold:   SetValves (3, 0, 0, 0)
      Wait (1000 ms)
Deliver_Reagent:
      Syringe:    SetFlowRate (10 ul/min)
      Syringe:    MoveTo (0 ul)
      WaitDone()
      Loop  Fill_Syringe      10
```

The first lines define the devices used in the script. Note that the devices are defined by *name* rather than *address*, so the names of all devices must be unique.

Below these declarations, each step is listed in order. For clarity, all state changes within each step are indented below the step name.

When learning to edit the text it may prove helpful to create a script using the Wizard, then to edit the generated text as the basis for your new script.

Most standard text editing functions are available, including **Cut**, **Copy** and **Paste**.

Script Errors

uProcess will review the script as you create or edit it and will indicate errors as follows:

If the script has an error the **Run** button will be grayed out and the script cannot be run.

The **Script Parsing** tab of the **Output** window will display the message

The script cannot be run until the script errors are resolved

The **Script Parsing** tab of the **Output** window will also list the line number and specific error. Double-clicking on the error will highlight the line with the error.

The step with the error will show a “broken” icon in the Sequencing Script list.

In the example shown in Figure 27, line 36 has an error. Device ‘water’ is not found, because the device was initialized as ‘Water’.

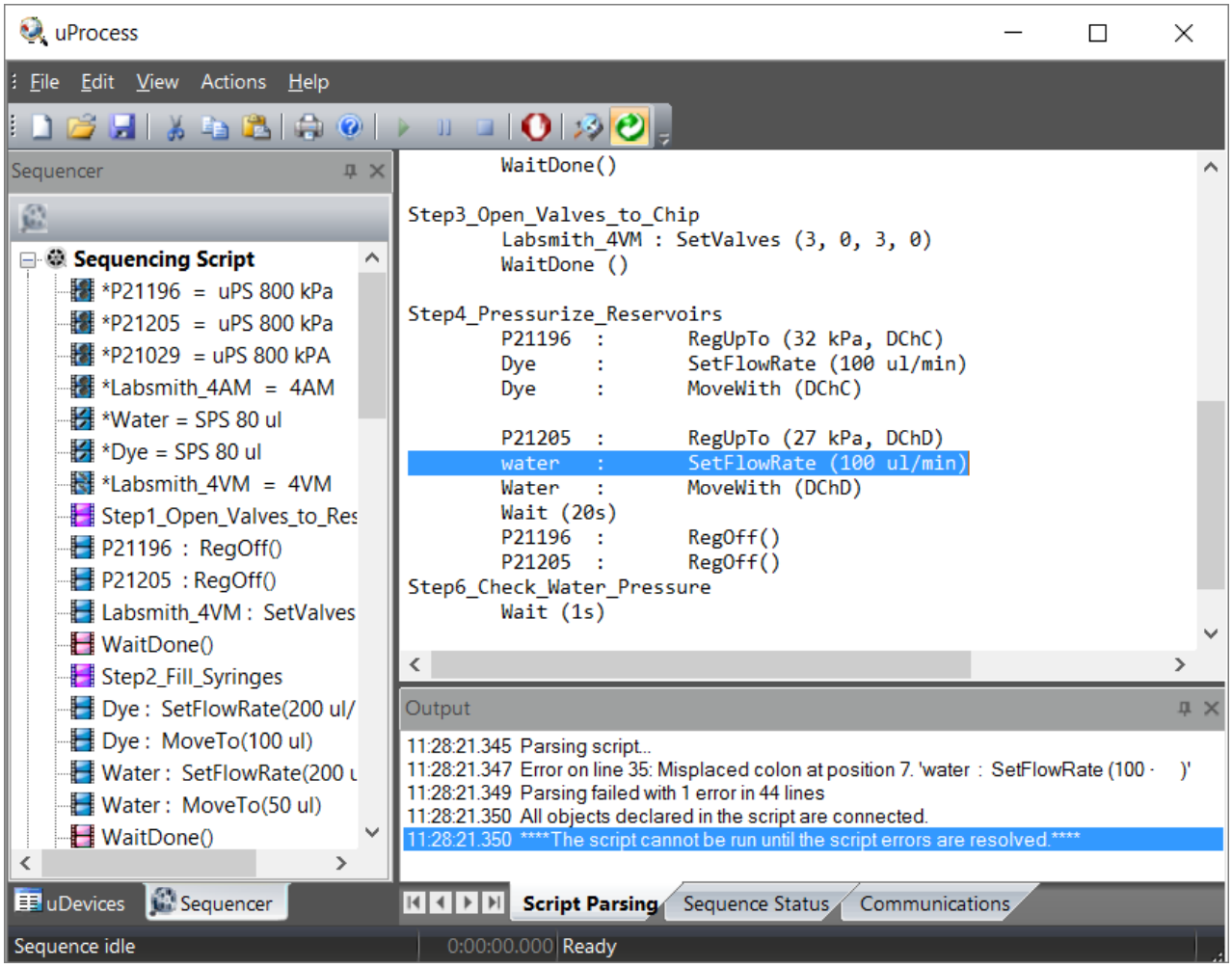


Figure 27. Sample script showing errors in line.

Missing Devices

All of the devices referenced in a process must be available for the process to run. If you open or create a script which calls out a device that is not present, error messages will appear in the ***uDevices Windows***, the **Script Parsing** tab of the ***Output Window***, and the ***Main Window*** (***Figure 28***).

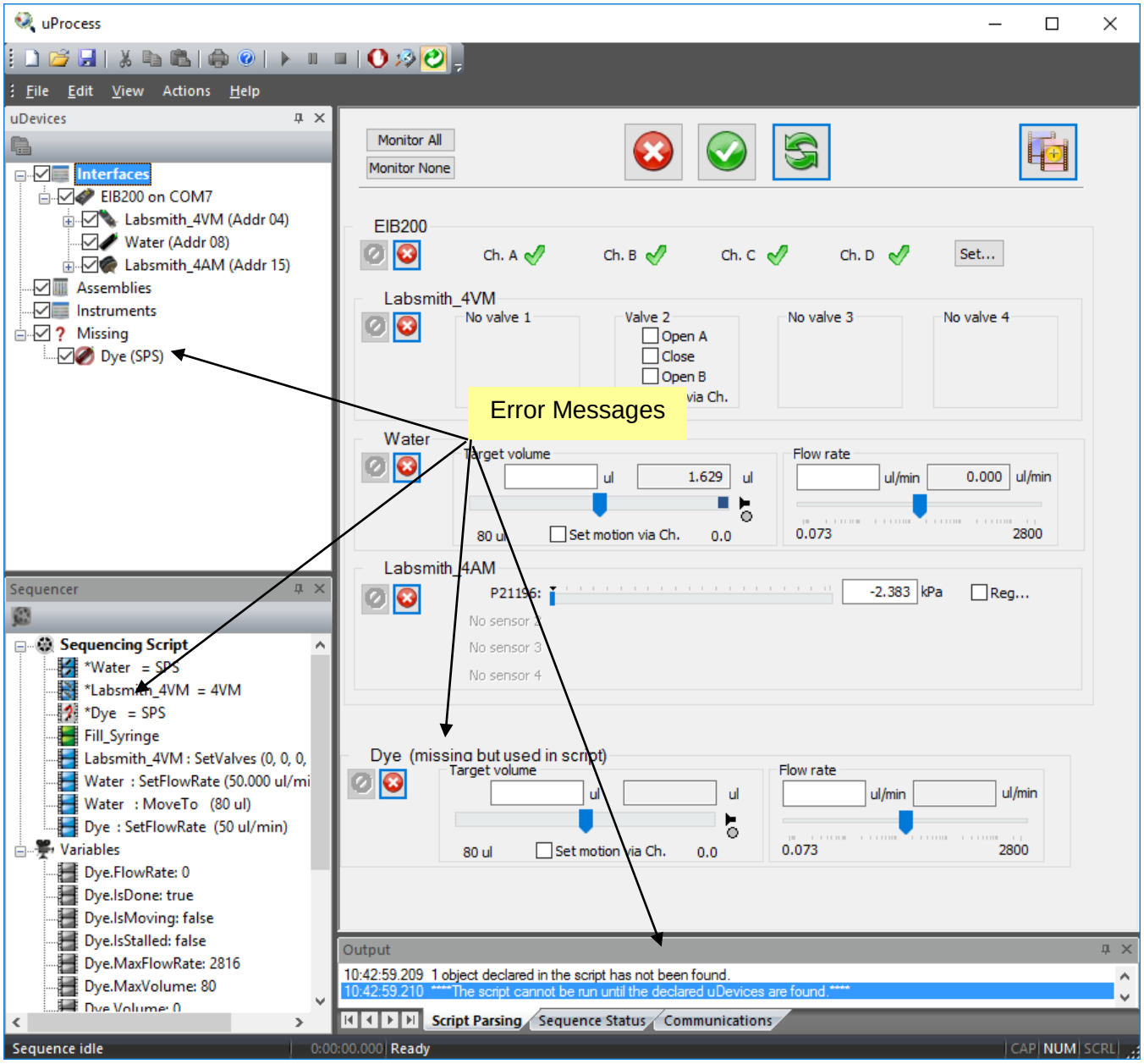


Figure 28. Error Message: Referencing a missing device.

For this example, the script references “Dye”, but the syringe is named “Water”. This error can be resolved by renaming the device (see ***Device Naming and Numbering***) or changing the script. To change the script:

Right click on “Dye” under the Missing devices tab and select **Rename …**

Change the name to “Water” and click **OK**.

Example Scripts

In the example shown in Figure 27, syringe pump is attached to an automated valve, a supply reservoir and a downstream process. The syringe will first pull fluid from a reservoir connected to the “A” port of the valve. The valve will then switch to the “B” position, and the syringe will deliver the fluid to the downstream process.

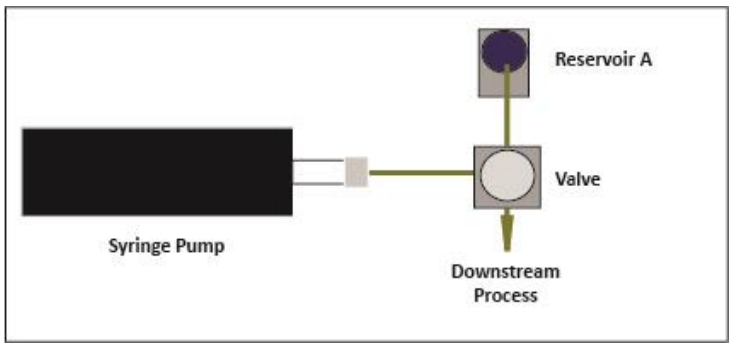


Figure 27. Hardware Setup for the Sample Process.

For this example the following devices are installed on one EIB:

- One 20µl syringe named “Reagent_Syringe”
- One 4VM01 manifold named “Reagent_Manifold”, with one valve connected at position 1

Rename the 20µ syringe to “Reagent_Syringe” and the 4VM01 manifold to “Reagent_Manifold” by right clicking on the device icon in the Devices Window and choose ‘Rename’.

In general, the name may have a maximum of 16 characters and may not include ‘;’, ‘<’, ‘<’, ‘*’, or ‘=’. Names should not include spaces or tabs. Names are case sensitive.

The automated script is created as follows:

In the uDevices Window select Interfaces to display the **Device Control Overview Window** (Figure 28).

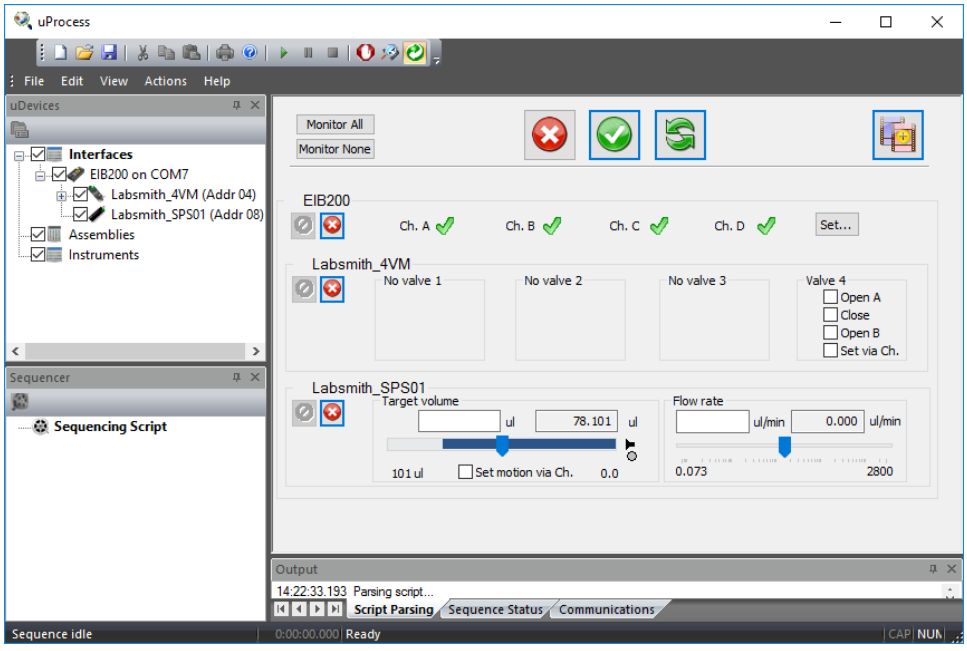


Figure 28. Installed Devices Shown in Device Control Overview Window.

Set the position for Valve 1 on “Reagent_Manifold” to **Open A.**

Depress the **Do Not Change uDevice Settings** button next to the Reagent Syringe.

Click the **Save As a New Sequence Step** button.

Name the new step “Open_Reservoir”.

Select ‘Wait until all devices are finished’ and ‘Continue’.

Click **OK** to add this step to the script and go on to the next step.

Set the **Target Volume** for the syringe named “Reagent_Syringe” to 10 µl and the Flow Rate to 50 µl/min (Figure 29).

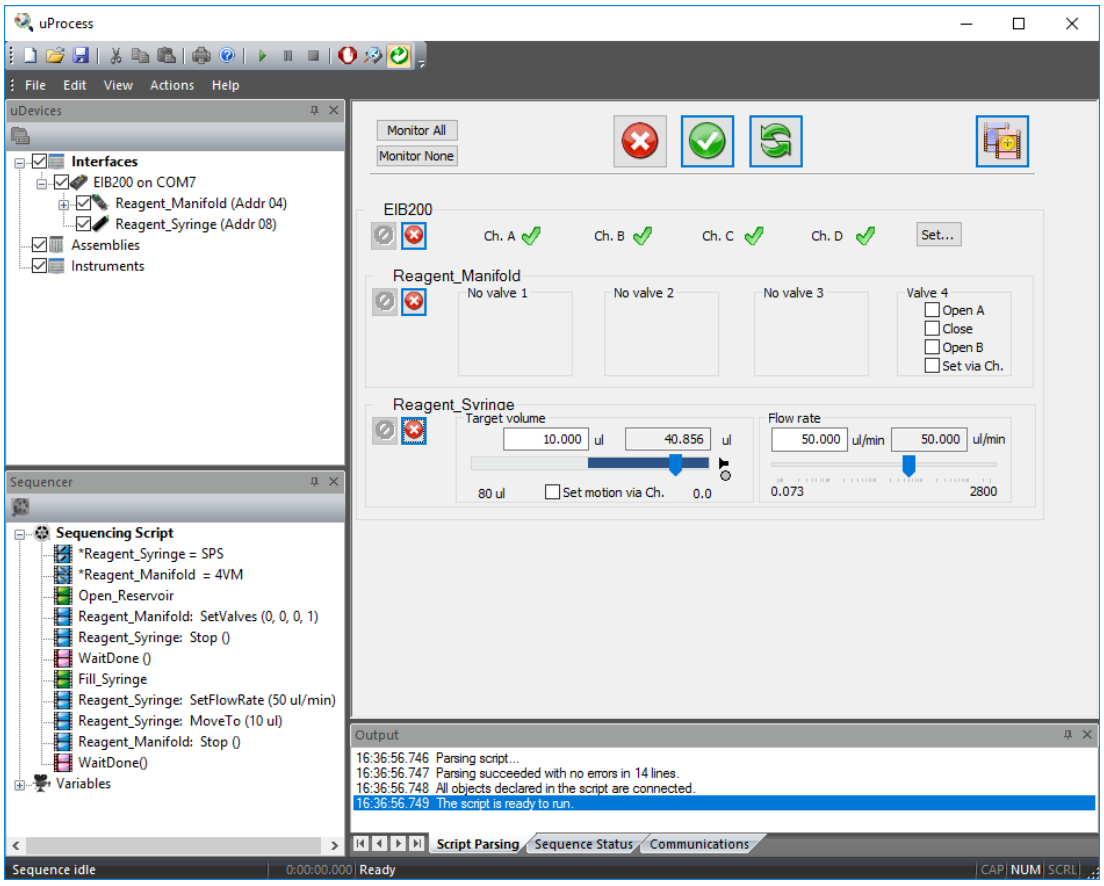


Figure 29. Step “Fill_Syringe” Settings.

Depress the **Do Not Change uDevice Settings** button next to the Reagent Manifold.

Click the **Save As a New Sequence Step** button.

Name the new step “Fill_Syringe”.

Choose ‘Wait until all devices are finished’ and ‘Continue’ (Figure 30).

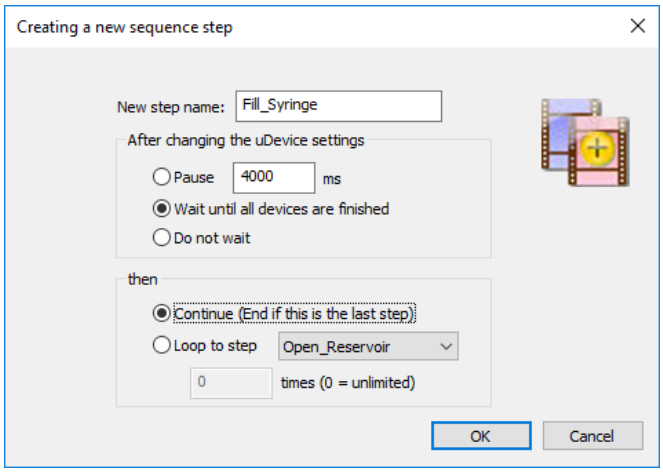


Figure 30. Step 1 Settings.

Click **OK** to add this step to the script. To view the process thus far, click the ***Sequencer Window*** tab then click on any step in the sequencing script (Figure 31). Select **Interfaces** in the ***uDevices Window*** to return to the **Device Control Overview Window** and continue building the script.

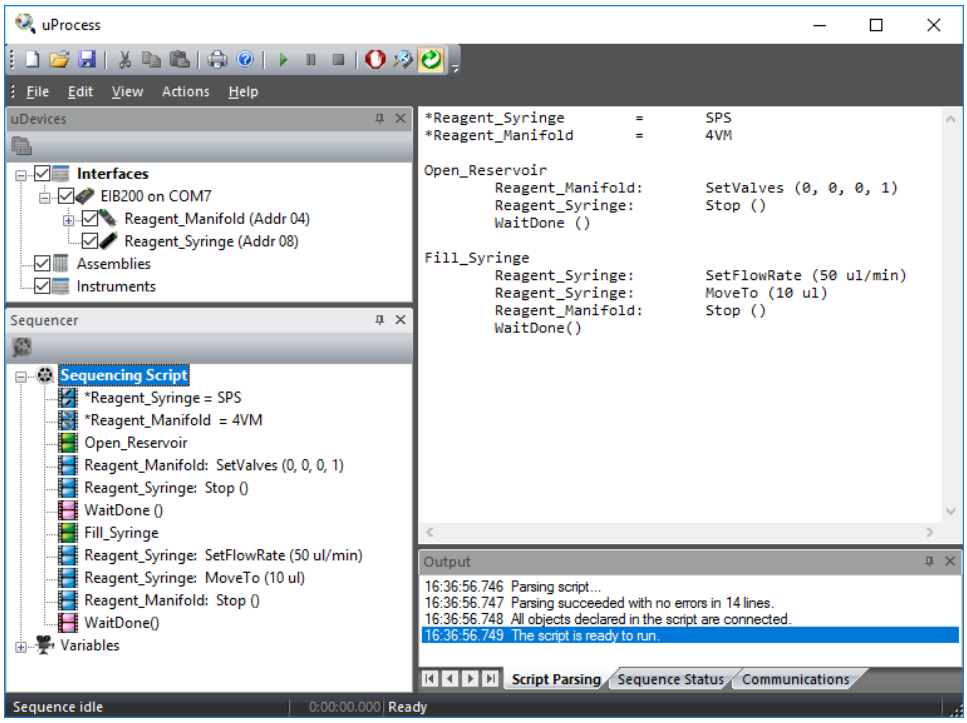


Figure 31. Step 1 Shown in the Sequence Tab and Main Window.

Set the position of **Valve 1** to **Open B** and depress the button for the Reagent Syringe**.**

Click the **Save As a New Sequence Step** button and name the new step “Open_Process”.

Choose **Pause** and set the time for 1 s (1000 ms). Choose **Continue** and select **OK** to go on to the next step.

Set the **Target Volume** to 0 and the **Flow Rate** to 10ul/min and depress the button for the Reagent Manifold.

Click the Save as a New Sequence Step button.

Name the step “Deliver_Reagent”.

Select ‘Wait until all devices have finished’.

Select **Loop to Step** and choose **Open_Reservoir** from the drop down menu. Set the loop count to 10, and the process will repeat for 10 iterations. (Figure 32).

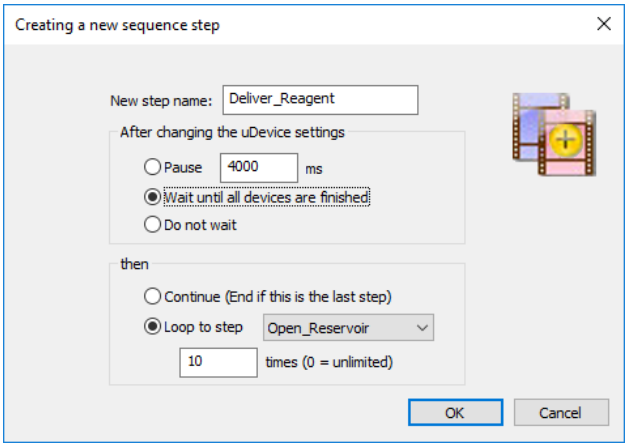


Figure 32. Deliver_Reagent Step Settings.

Click OK. Again, select the Sequencer Window tab and click on a step in the script to view the complete script (Figure 33).

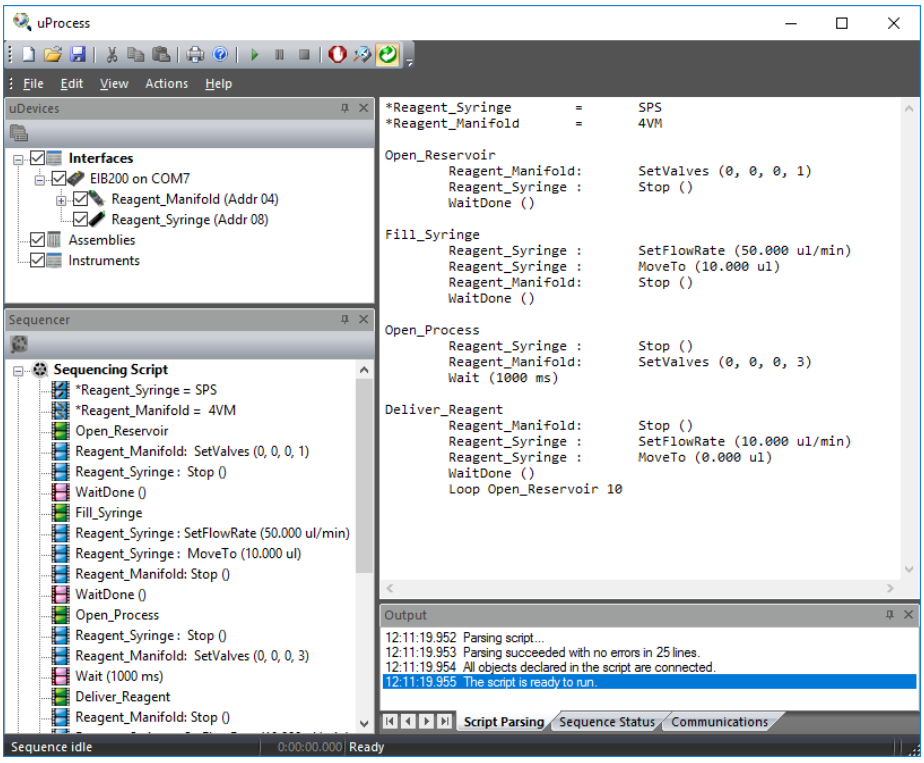


Figure 33. The Complete Example Script.

Press the **Run** button to initiate the automated script.

When a script is running, the current step will be highlighted in the ***Main Window*** to allow tracking of the script progress. The status of the devices and the step progression can be monitored real time. As shown in the figure below, the ***Device Control Overview Window*** shows the real time positions of the syringes and valves; the **Sequence Status** tab of the ***Output Window*** shows which step is currently running and the number of iterations which have been completed. The bottom left corner of the window shows how long the script has been running (Figure 34).

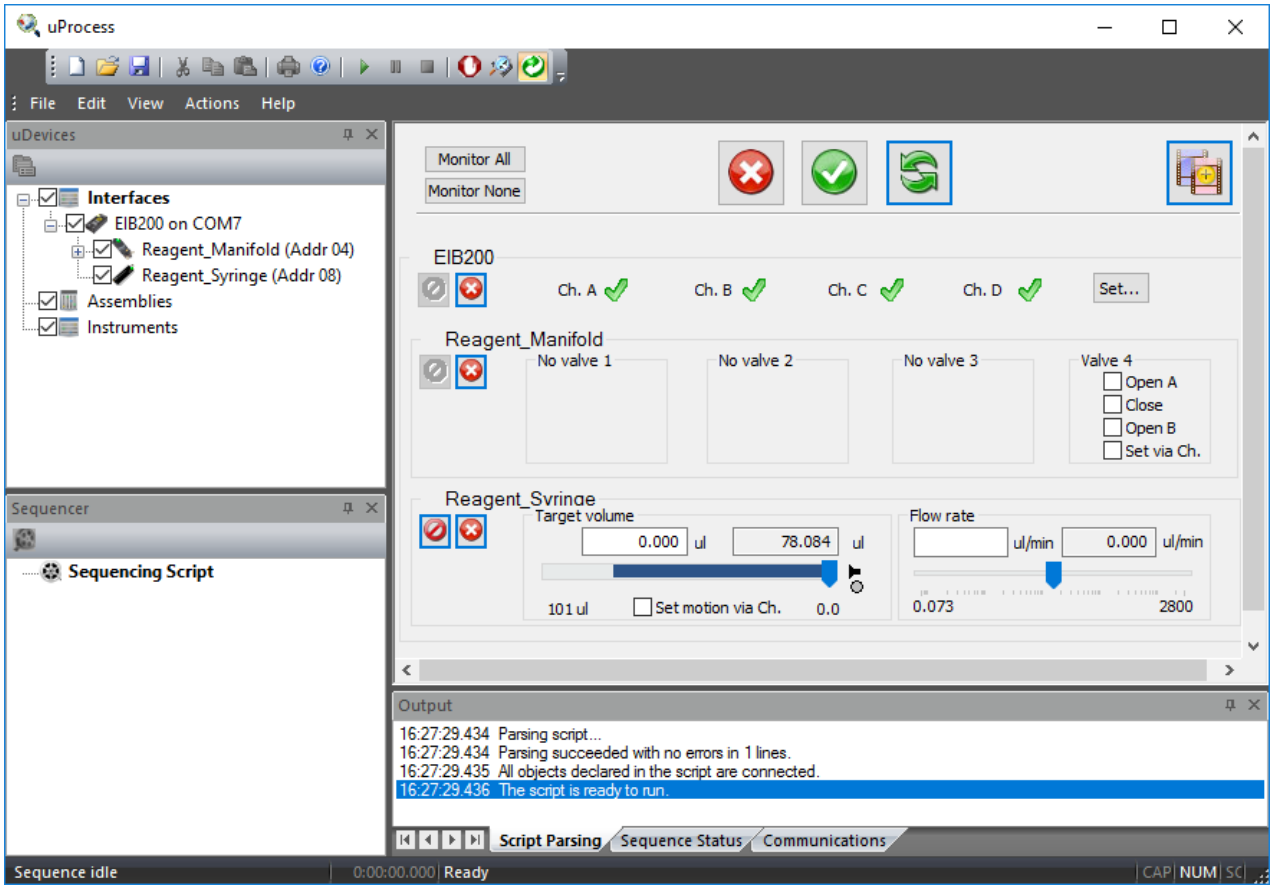


Figure 34. Sequence Status Tab

Syringe Pump Volume vs Delivery Time

In the above example the syringe is set to deliver 10 µl of reagent at 10 µl/min. In order not to overshoot the delivery of the reagent, the syringe will slow down just before reaching the target volume; therefore, the delivery time may be more than 1 minute. This is most pronounced when delivering at high flow rates.

For some processes the infuse time may be more critical than the total delivery volume. For this case you can use the **Wait()** and **Stop()** functions together to set the infuse time. The script from the above example is shown below.

In this script, the syringe is called “Syringe” and the 4VM01 manifold is called “Manifold”.

```
*Syringe    =     SPS01 20 ul
*Manifold   =     4VM01
Open_Reservoir
      Manifold:   SetValves (1, 0, 0, 0)
      WaitDone()
Fill_Syringe
      Syringe:    SetFlowRate (50 ul/min)
Syringe:    MoveTo (10 ul)
      WaitDone()
Open_Process
      Manifold:   SetValves (3, 0, 0, 0)
      Wait (1000 ms)
Deliver_Reagent
      Syringe:    SetFlowRate (10 ul/min)
      Syringe:    MoveTo (0 ul)
      WaitDone()
      Loop  Fill_Syringe      10
```

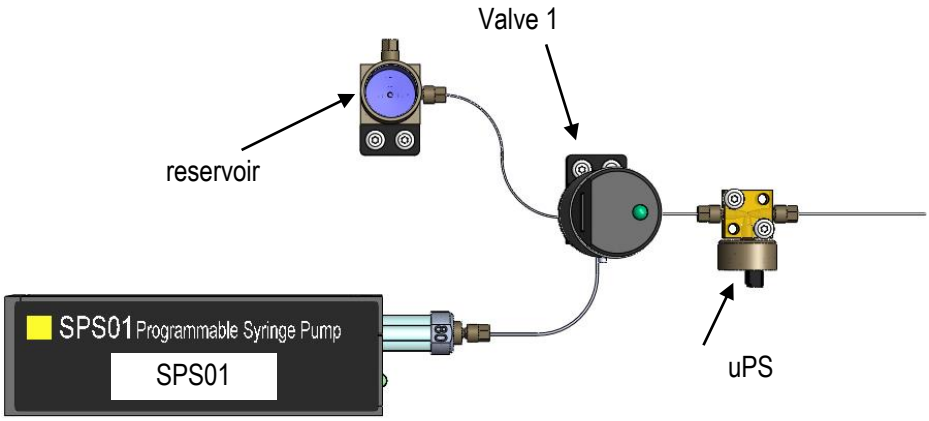
The total volume delivered is 10 µl, which at 10 µl/min should take approximately 1 minute. If a delivery time of exactly 1 minute is desired, the script could be modified as follows (changes are in bold):

```
*Syringe    =     SPS01 20 ul
*Manifold   =     4VM01
Open_Reservoir
      Manifold:   SetValves (1, 0, 0, 0)
      WaitDone()
Fill_Syringe
      Syringe:    SetFlowRate (50 ul/min)
      Syringe:    MoveTo (11 ul)    ; provide extra volume so the
                                    ; syringe won't stop prematurely
      WaitDone()
Open_Process
      Manifold:   SetValves (3, 0, 0, 0)
      Wait (1000 ms)
Deliver_Reagent
      Syringe:    SetFlowRate (10 ul/min)
      Syringe:    MoveTo (0 ul)
      ;WaitDone()  this command is commented out
      Wait (60 s)             ; allow the syringe to run for 60 s
      Syringe:    Stop()      ; stop the syringe
      Loop        Fill_Syringe      10
```

In the second example, the syringe is instructed to stop after exactly 60 seconds, which will be after approximately 10 µl is delivered.

Pressure Regulation Example Scripts

The uPS pressure sensors and SPS01 syringe pumps can be used together with the uProcess software to regulate the pressure in a device or create pressure driven flow. In the simple pressurization example below, the SPS01 syringe pump is filled from the reservoir, and then it is used to pressurize a device. When the pressure reaches 100 kPa the script is complete.



Simple pressure regulation application (uPB-05 breadboard, 4VM manifold, and 4AM manifold not shown for clarity)

In order to rename the pressure sensors, click on **4AM** in the uDevices window under **Interfaces**.
In the main window, click on 'Configuration'. This will open up a new dialogue box.
Click on 'Sensor 1' to edit the name of the first sensor. This will open another dialogue box.
In the box under 'Sensor Name' put in the new name. In this script, the sensor is called 'uPS'.
Remember the name is case sensitive.

Simple Pressure Regulation Example Script

```
*SPS01      =     SPS01 80 ul
*uPS        =     uPS01 800 kPa
*4AM        =     4AM01
*4VM        =     4VM01

Switch_Fill:      ; switch valve to “fill” position
      4VM:        SetValves (1, 0, 0, 0)
      4VM:        WaitDone()

Fill:             ; fill syringe with liquid from reservoir
      SPS01:      SetFlowRate (150 ul/min)
      SPS01:      MoveTo (80 ul)
      SPS01:      WaitDone()

Switch_Dispense:  ; switch valve to “dispense” position
      4VM:        SetValves (3, 0, 0, 0)
      4VM:        WaitDone()

Pressurize:
      uPS:        RegUpTo (2 kPa)
      SPS01:      SetFlowRate (20 ul/min)
      SPS01:      MoveTo (0 ul)
      uPS:        WaitDone()
      SPS01:      Stop()      ; stop the syringe pump
      4AM:        Stop()      ; stop the regulation
```

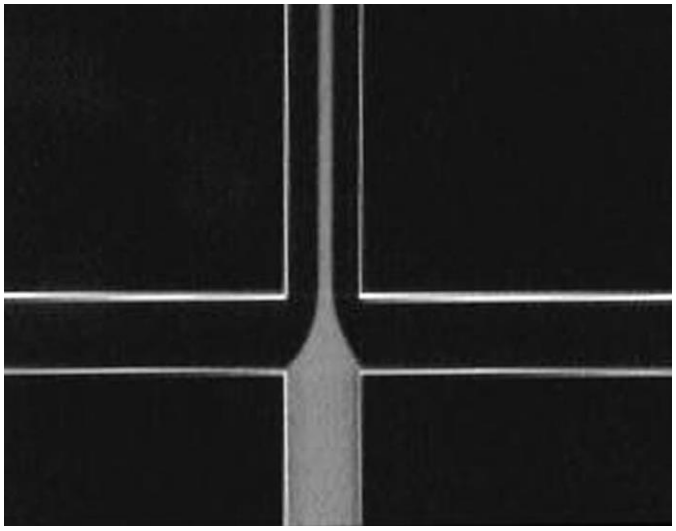
One potential issue with the above script is if the syringe reaches the end before the pressure reaches 2 kPa the script will never complete. To account for this condition, the last step (“Pressurize”) can be changed to include a loop that looks to see whether the syringe has finished. If the syringe finishes before the pressure is reached, the valve switches back to the reservoir to refill the syringe and then the device is pressurized again. This script is shown below.

Pressure Regulation Example Script with auto re-fill of syringe if pressure is not reached

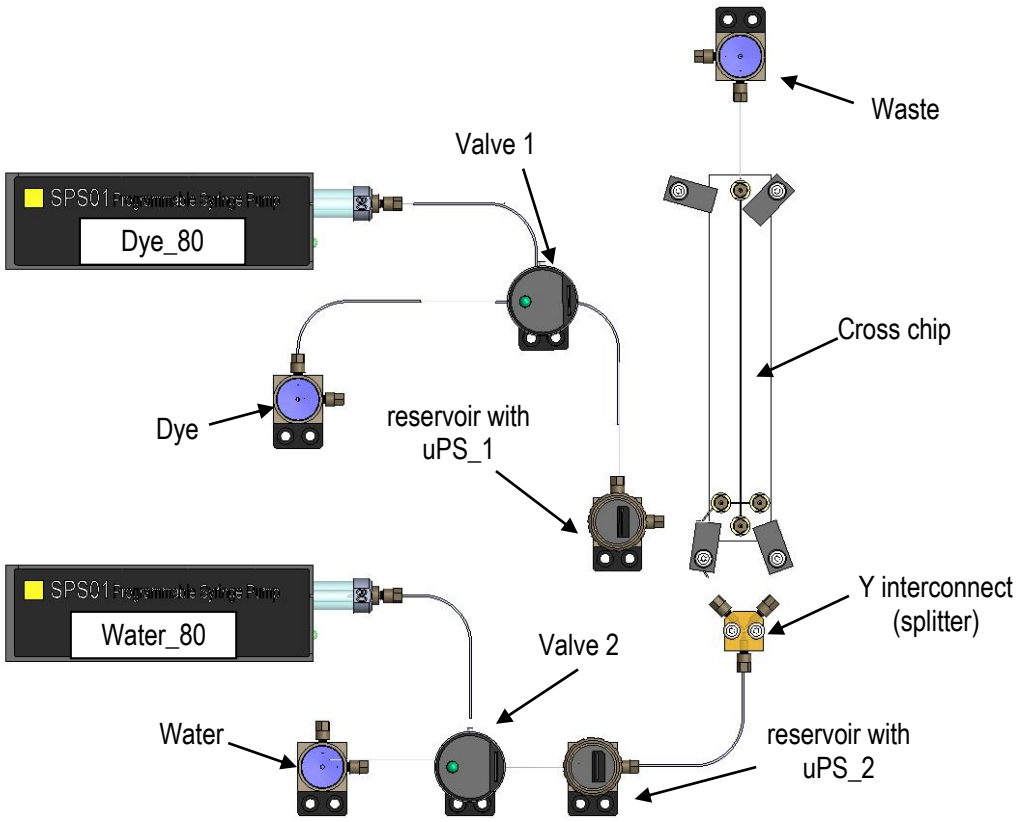
```
*SPS01      =     SPS01 80 ul
*uPS        =     uPS01 800 kPa
*4AM        =     4AM01
*4VM        =     4VM01

Switch_Fill ; switch valve to “fill” position
      4VM:        SetValves (1, 0, 0, 0)
      4VM:        WaitDone()

Fill              ; fill syringe with liquid from reservoir
      SPS01:      SetFlowRate (150 ul/min)
      SPS01:      MoveTo (80 ul)
      SPS01:      WaitDone()

Switch_Dispense   ; switch valve to “dispense” position
      4VM:        SetValves (3, 0, 0, 0)
      4VM:        WaitDone()

Pressurize
      uPS:        RegUpTo (100 kPa)
      SPS01:      SetFlowRate (20 ul/min)
      SPS01:      MoveTo (0 ul)

Check_Loop
      SPS01:      ifdone()
      Goto        Switch_Fill
      uPS:        ifdone()
      Goto        Finish
      Wait(1 s)
      Goto        Check_Loop

Finish
      SPS01:      Stop()      ; stop the syringe pump
      4AM:        Stop()      ; stop the regulation
      Beep()                  ; indicate script is complete
```

Pinched Flow Example Script with Pressure Regulation

The uProcess system can also be used to create near-constant pressure flow, as shown in the following pinched flow application example. Fluorescent dye is injected in the main leg of a cross chip and water is injected in the two cross arms to create pinched flow across the width of the channel. A reservoir (BB_RES) out fitted with an o-ring and cap is placed upstream of each chip inlet. Each reservoir is partially filled with the corresponding liquid (dye or water), leaving an air gap at the top to allow flow dampening. A uPS pressure sensor is installed in the cap of the reservoir to monitor and regulate the pressure.



Pinched Flow



Pressure regulated hydrodynamic flow application (iBB breadboard, 4VM manifold, and 4AM manifold not shown for clarity)

The script below will:

1. Switch_Fill: open the valves to the open reservoirs;
2. Fill: fill the syringes;
3. Switch_Dispense: open the valves to the chip;

4. Regulate: dispense the syringe fluids toward the chip
   DYE
   a. Dye_80 syringe dispenses until reservoir_1 pressure reaches 46 kPa
   b. Dye-80 syringe stops, pressure monitored until pressure drops below 45 kPa
   c. water dispense until reservoir_2 pressure reaches 37.5 kPa

```
*uPS_1 =          uPS01 800 kPa
*uPS_2 =          uPS01 800 kPa
*Dye_80=          SPS01 80 ul
*Water_80 = SPS01 80 ul
*4AM =            4AM01
*4VM =            4VM01

Switch_Fill:      ; switches valves to “fill” position
      4VM:        SetValves (3, 3, 0, 0)
      4VM:        WaitDone()
Fill:             ; fills syringes with liquid from reservoirs
      Water_80:         SetFlowRate (150 ul/min)
      Water_80:         MoveTo (100 ul)
      Dye_80:           SetFlowRate (150 ul/min)
      Dye_80:           MoveTo (100 ul)
      Water_80:         WaitDone()
      Dye_80:           WaitDone()

Switch_Dispense:  ; switches valves to “dispense” position
      4VM:              SetValves (1, 1, 0, 0)
      4VM:              WaitDone()

Regulate:         ; dispense the syringes to pressurize the
                  ; reservoirs, when the pressure reaches max set
                  ; value, syringes will stop pumping & wait until
                  ; the pressure has dropped below min value
      Dye_80:           SetFlowRate (100 ul/min)
      uPS_1:            HysteresisRegTo (45 kPa, 46 kPa, DChA)
      Dye_80:           MoveWith (DChA)
      Water_80:         SetFlowRate (100 ul/min)
      uPS_2:            HysteresisRegTo (37 kPa, 37.5 kPa, DChB)
      Water_80:         MoveWith (DChB)

CheckRegulation:  ; loop that checks if either syringe is empty (at
                  ; 0) and switches to refill if so
      Water_80:   ifdone()
      Goto        Switch_Fill
      Dye_80:     ifdone()
      Goto        Switch_Fill
      Wait (1 s)
      Goto        CheckRegulation
```

Constant-pressure pinched flow example script.

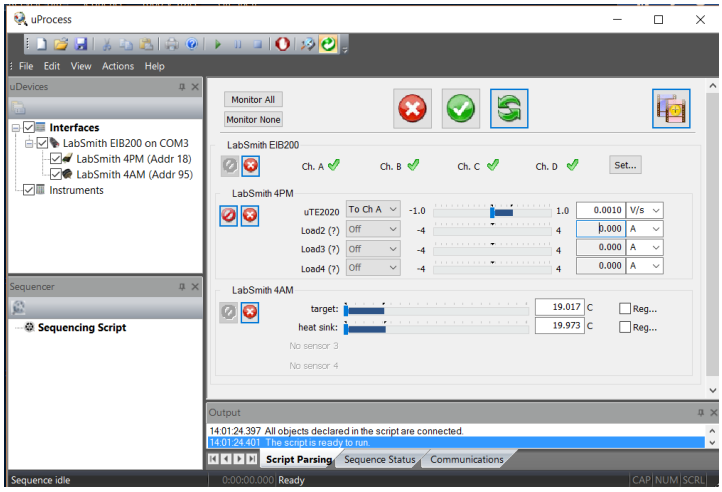
The above script will run continuously until it is manually stopped.

Thermal Application Example: Switching Between Two Set Temperatures

This example shows how to use the **uTE controller** and **uTS sensor** to rapidly switch between two temperature settings, 90 °C and 50 °C, at 5 minute intervals.

A voltage ramp rate of 0.01 V/S is used to optimize temperature ramp rate rather than precision.

1) Click on *Interfaces* in the uDevices tree to view the uDevice™ window.

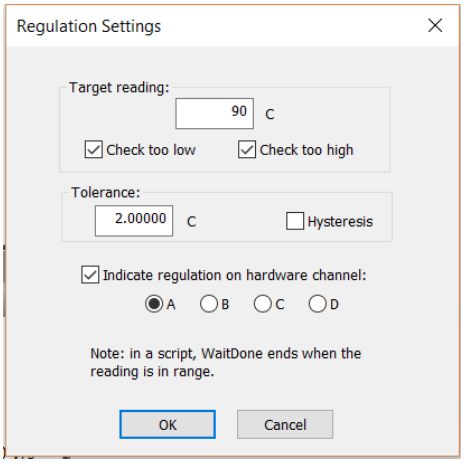


2) Using the drop-down menu, set the **uTE Peltier** load to “To Ch A.”

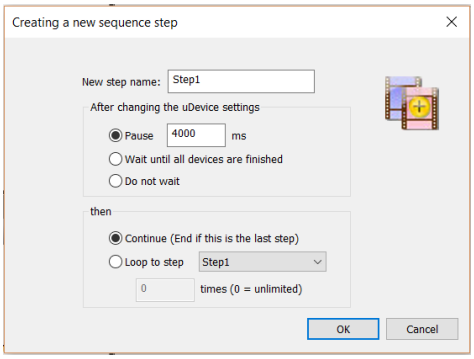
3) Set the uTE voltage ramp rate to .01 V/s.

4) Click on the *Reg…* box next to the target temperature listing to bring up the Regulation Settings window.

5) Set the target reading to 90 °C and click OK.

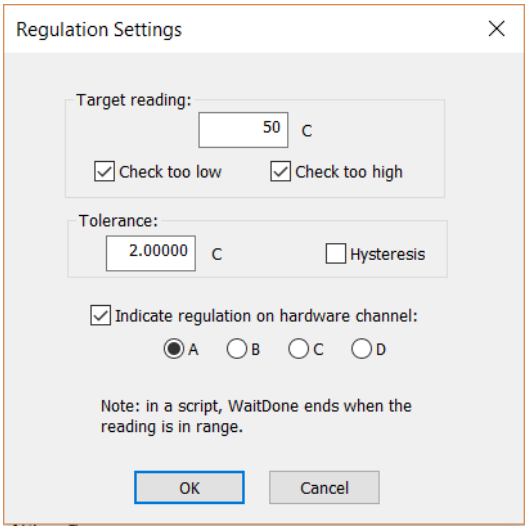


6) Click on *Save as New Sequence Step* ( ) and change the setting to *Pause*. Leave the time at 4000 ms (this will be changed directly in the script). Click OK to save the settings.

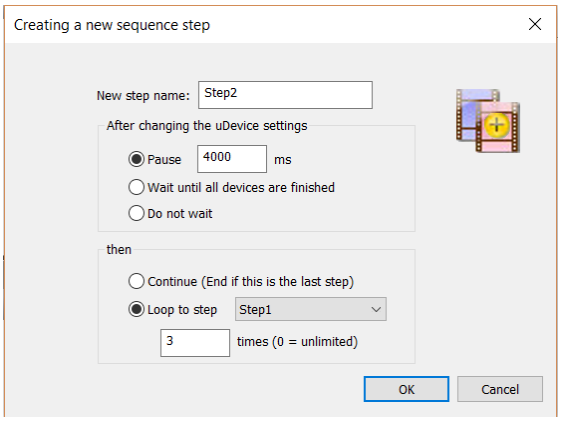


7) Click on the *Reg…* box next to the target temperature listing to bring up the Regulation Settings window.

8) Change the Target reading to 50 °C and click OK.



9) Click on *Save as New Sequence Step* ( ). Set the pause time to 4000 ms and select *Loop to Step*. Change the loop to *Step1* for 3 times click OK.



10) Select *Sequencing Script* to view the script. The script should look similar to the following.

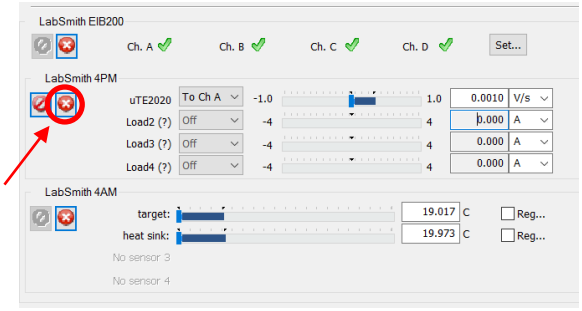
```
*target           =     uTS01
*LabSmith_4AM     =     4AM01
*LabSmith_4PM     =     4PM01
Step1:
      LabSmith_4PM:     SetPerChannel( 1, DChA, 0.01000 V/s)
      target:     RegBetween (88.000 C, 92.000 C, DChA)
      Wait (4000 ms)

Step2:
      LabSmith_4PM:     SetPerChannel (1, DChA, 0.01000 V/s)
      target:     RegBetween (48.000 C, 52.000 C, DChA)
      Wait (4000 ms)
      Loop   Step1      3
```

The script will start to regulate the target temperature to 90 °C. After 4 seconds it will start to regulate down to 50 °C. This cycle will be repeated for a total of 3 iterations. At this point, the script will be complete, but the **uTE** will continue to regulate to 50 °C (the last setting) until the script is manually stopped by pressing *Stop All uDevices* .

Two changes will improve the script:

- An end condition added to turn off the regulation after the script is done.
- The pause time can be adjusted to 5 minutes. The pause clock will not start until the target temperature is in range.

11) To add the end condition, click on *Stop this Device* located in the **4PM** panel.



Then, click on *Save as New Sequence Step* ( ) before clicking OK to create Step3 .

12) Adjust the pause time by clicking *Sequencing Script* to bring up the sequencing script control window. Delete '4000 ms' and type '5 min' in both Step1 and Step2.

13) Adding a WaitDone() command before the Wait() command will delay the start of the Wait() timer until the target temperature reaches the specified temperature range.

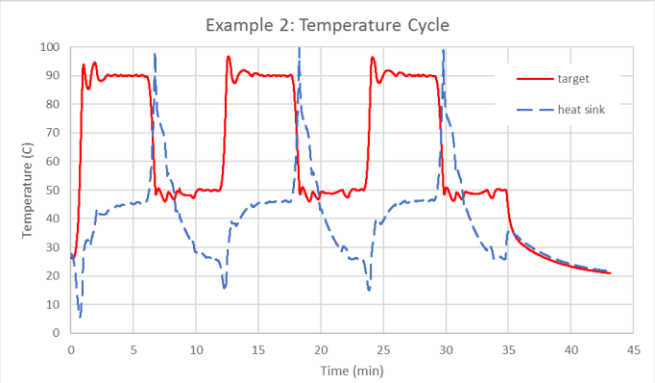
14) The sequencing script should now match the following (items changed are in bold).

```
*target           =     uTS01
*LabSmith_4AM     =     4AM01
*LabSmith_4PM     =     4PM01
Step1:
      LabSmith_4PM:     SetPerChannel (1, DChA, 0.01000 V/s)
      target:     RegBetween (88.000 C, 92.000 C, DChA)
      WaitDone()
      Wait (5 min)

Step2:
      LabSmith_4PM:     SetPerChannel (1, DChA, 0.01000 V/s)
      target:     RegBetween (48.000 C, 52.000 C, DChA)
      WaitDone()
      Wait (5 min)
      Loop  Step1 3

Step3:
      LabSmith_4PM: Stop()
      WaitDone()
```

15) Go back to the uDevice™ window (click on *Interfaces* in uDevice™ tree) to monitor temperatures before starting the script.

16) To log the temperature data, click on *File>Log Status/Measurements…* to open a log file. uProcess will prompt you to name the file, unless uProcess is set to autoname the file in *View>Status Meas Logging Options…*

17) Run the sequence program ( ). The graph below shows the dataplot obtained using this example code.



CHAPTER 6. SVM340 AUTOMATION AND CONTROL

uProcess software versions 1.28 and higher allow control of the SVM340 within the uProcess software. The instructions below describe how to use uProcess to connect to and control the SVM340. This guide assumes you have basic operational knowledge of the SVM340 operation and uScope software. Please refer to the full uScope user's manual for detailed SVM340 instrument and software instructions.

Software Requirements

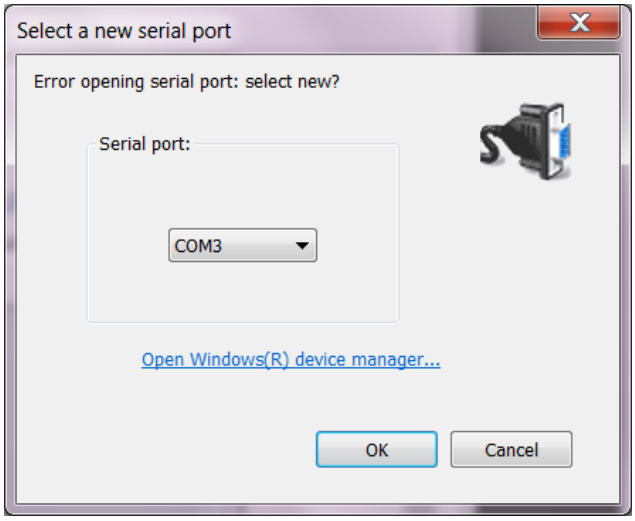
You must have the following software installed on your computer to use the SVM340 in uProcess:

uScope version 1.45 or later
uProcess version 1.28 or later

After downloading the latest uScope software, update the SVM firmware by going to SVM>Update firmware>Update all.

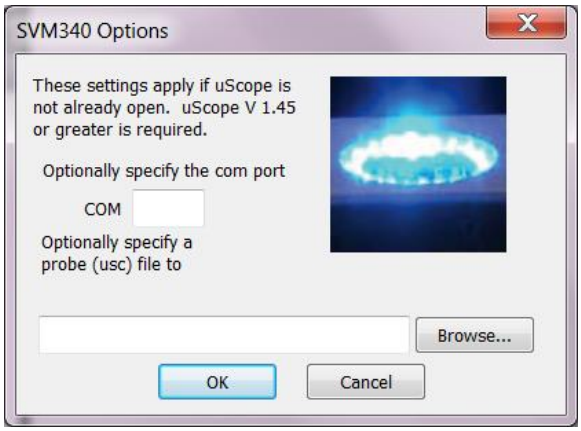
Connecting to the SVM340 in uProcess

1. Start the uProcess software. If you would like to connect to uProcess equipment via an EIB you may do so at this time.
2. Turn on the SVM340 and make sure the RS232 communication cable is connected to the computer (either to a RS232 port or to a USB port through a RS232-USB adapter).
3. Start the uScope software.
4. If the serial port is not found automatically the "Select a new serial port" window will open. If this serial port window does not open, the correct com port has been found and you can skip to step 6.

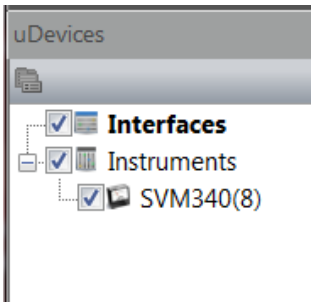


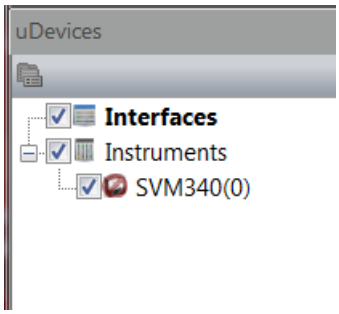
5. Select the correct serial port from the dropdown list and click "OK" (refer to the SVM user's manual if the correct serial port cannot be found).
6. Once the correct serial port has been found, you will select the optics module. If the window to open a video file automatically opens, click "Cancel" to get to the uScope main window.
7. Select Video>Video Capture Device to select the installed optics module.
8. Once the SVM is operational you will return to the uProcess software.

9. If SVM340(0) is listed under instruments, right click on the name and select “Remove this Instrument”.
10. Right click on ‘Instruments’ in the uDevices tree and select “New Instrument>SVM340…” to open a SVM portal.



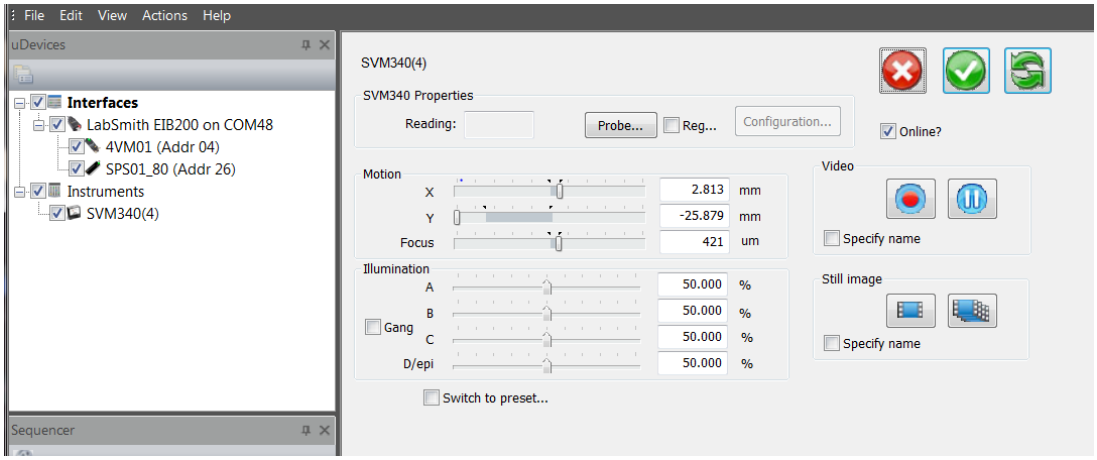
11. Leave the COM port blank.
12. Click ‘OK’
13. If the uDevices window shows SVM340(#) and the number is non-zero, you are connected to the SVM, go to the next section “Using the SVM in uProcess”.
14. If the SVM340 Icon has a red circle/slash through it and the number is zero (as shown on the right) you are not connected to the SVM. In this case, right click on the SVM icon and select ‘Remove this Instrument”, close the uProcess and uScope software, and start again on step 1. Contact LabSmith to if you still cannot connect.





Using the SVM340 in uProcess

1. The SVM controls are viewed one of two ways. Each view has slightly different control options.
    a. Click on the 'Interfaces' icon in the uDevices tree to show all devices connected (via EIB or external instruments).
    b. Click on the SVM340 icon in the Instruments tree to show only the SVM340 controls.



2. To alter the X, Y or focus setting
    a. If using the Interfaces window, first place a check mark beside the setting you want to change (not required in the SVM solo window)
    b. Move the slider or type in the desired position
    c. Press the 'Apply All Settings Now' button
    d. Or, press the 'Update all uDevices Continuously' button to continuously update the settings
3. To alter the Illumination setting
    a. In the Interfaces window, click on 'Change Illumination…' and slide the channel bars to the desired value and click 'OK'.
    b. if the 'Continuously Update' button is not depressed, you will need to press the 'Apply All Settings' button for the settings to be changed.
4. Preset values must be set using the front panel (for presets A-D), or from the uScope software (A-J). Once they are set, the can be recalled from the SVM340 portal in the uProcess software.
5. To take a Snapshot Image
    a. Leave the box next to 'Name' blank. The image name will use the protocol specified in uScope (in uScope, go to File>Snap Shot File Saving).
    b. The snapshot will use the setting specified in uScope under View>Snap-shot Settings…

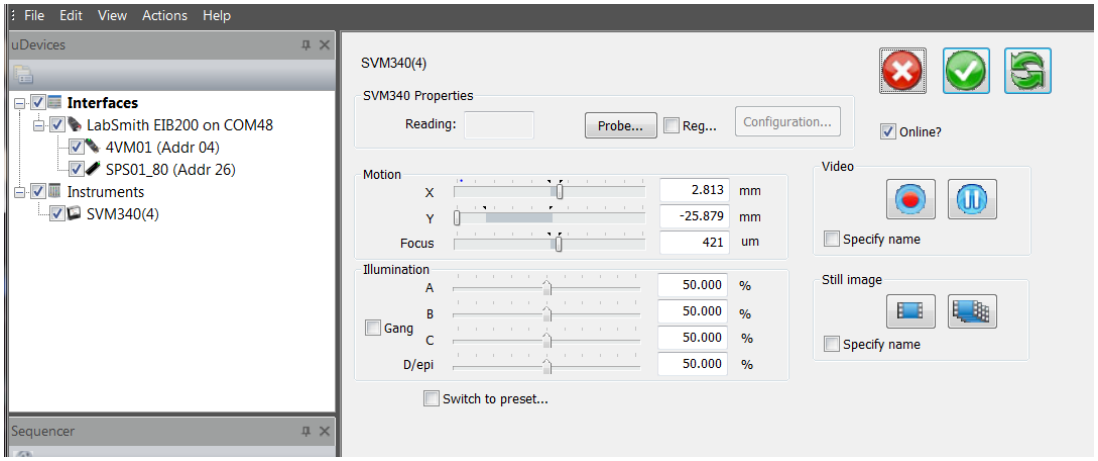
c. Press to save a snap shot.

d. If the 'Continuously Update' button is not depressed, you will also need to press the 'Apply All Settings' button to take the snapshot.

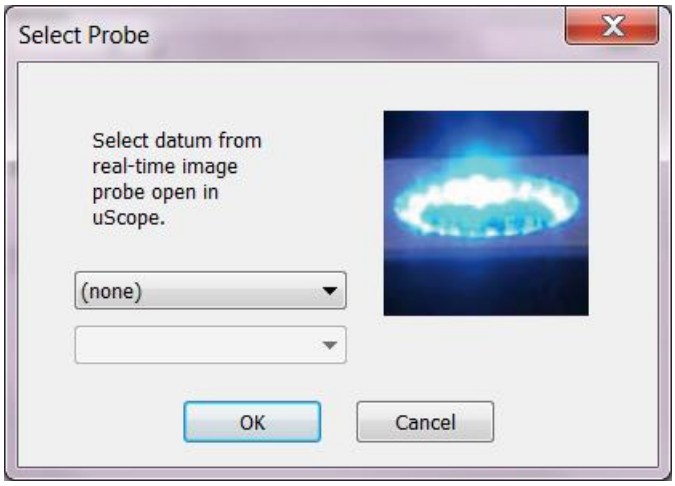
e. Depending on the Snap Shot File Saving settings, you may be prompted for a file name after each snapshot (unless Auto-name files is selected)

f. When you are done taking snapshots, you will need to press the 'Do not Change uDevices button to exit snapshot mode.

6. To take a Time-averaged Image
   a. Follow the instructions above for Snap-shot image. However, the image will be time averaged until the button is pressed for a second time. Again, if the 'Continuously Update' button is not activated, you will also need to press the 'Apply All Settings' button before both starting and stopping the time-averaged image.
7. To take a Video
   a. Leave the box next to 'Name' blank. The video name will use the protocol specified in uScope (in uScope, go to File>Video File Saving).
   b. The video will use the setting specified in uScope under View>Video capture Settings…
   c. Press to start recording a video. NOTE: if the 'Continuously Update' button is not activated, you will also need to press the 'Apply All Settings' button to start, pause, or stop the recording.
   d. Press to pause the recording (press again to unpause), and again to stop the recording.
   e. Depending on the Video File Saving settings, you may be prompted for a file name after each recording (unless Auto-name files is selected).
8. To monitor a probe:
   a. Note: your probes must first be created in the uScope software before they can be imported and monitored in uProcess. See the uScope user's manual for instructions on creating probes.
   b. Once your probe(s) have been created, click on the SVM340(#) icon in the device tree to bring up the SVM340 device window.



c. Click on ‘Probes…’ to open the probes window



d. Click on the dropdown bar to select the probe you would like to monitor in uProcess.

SVM340 Commands Available in the uProcess Sequencing Script

Note: This section assumes the user is familiar with the uProcess automated script functions. If not, please review Chapter 5 before continuing.

The uProcess script wizard can be used to automatically configure the script file and add many of the commands listed below. The SVM340 commands are added by manually adjusting the settings in the interface window and then clicking the "Save as New Sequence Step" button ( ).

The following header must be included in the scrip file when using SVM340 commands:

```
*SVM340(#)   =      SVM340
```

where # is the SVM340 com port number

All commands to the SVM340 will have format:

```
SVM340(#):  COMMAND
```

All available commands are listed below.

Video and Image Saving

NOTE: uProcess has the option to specify a file name in the script when saving an image (snap shot, time-averaged image, or video).

- If the file name is NOT specified, uProcess will use the naming configuration that has been set up in uScope (File>Snap Shot File Saving… or File>Video File Saving…)

- If the file name IS specified, a file name or address with spaces in it MUST use quote around the name. (see RecordImage below for example)

RecordImage < optional filename>	; takes a snapshot image
AverageImage On <optional filename>	; starts a time-averaged snapshot image
AverageImage Off	; stops a time-averaged image
RecordVideo On < optional filename>	; starts video recording
RecordVideo Off	; stops video recording
PauseVideo On	; pauses recording
PauseVideo Off	; resumes recording

Examples:

```
SVM340(3):   RecordVideo On
SVM340(3):   RecordVideo On C://video1.jpg
SVM340(3):   RecordVideo On "C://saved videos/video1.jpg"
```

Focus Stage Position and Illumination Adjustment

MoveToX {x} {units} ; moves to position x units from center

MoveToY {z} {units} ; moves to position y units from center

FocusTo {z} {units} ; moves to position z units from center

MoveTo {x] {units} {y] {units} {z] {units} ; moves to position x, y, z

MoveByX dx <units> ; moves dx units from current position

MoveByY dy <units> ; moves dy units from current position

FocusByZ dz <units> ; moves dz units from current position

Acceptable units: mm, um, in, cm, m

Examples:

SVM340(3): MoveToX -5 mm
SVM340(3): MoveByY -0.2 mm
SVM340(3): FocusTo 500 um
SVM340(3): MoveTo -5 mm 10 mm 500 um

SetLEDs {C1} {C2} {C3} {C4} ; sets illuminator channels intensity from off (0) to full (1.0)

Example: SVM340(3): SetLEDs 0.5 0.5 0.0 1.0

SwitchToPreset {1 to 10} ; moves to preset position/illumination setting, 1=A, 2=B, etc.

Example: SVM340(3): SwitchToPreset 2

Stop ; stop all motion

Example: SVM340(3): Stop

Regulation using Probe Data

SelectReading {ProbeName} {ProbeDetails} ; reads probe data for regulation

Use with one of the following regulation commands:

RegUpTo {value}
RegDownTo {value}
RegBetween {min_value max_value}

Examples

SVM340(4): SelectReading Polygon1 MeanGray
SVM340(4): RegBetween 70 130

SVM340(4): SelectReading MPixels0 I(3,0)
SVM340(4): RegDownTo 4000

APPENDIX A. MENUS, TOOLBARS, WINDOW CONTROLS

Menu Bar

The menus contain the following functions:

File New, Open, Save, or Print.

Edit Undo, Cut, Copy, or Paste text while editing an automated process.

View Choose which uProcess windows and toolbars are displayed.

Actions Run, Pause or Stop an automated process, control hardware communications, and update firmware and calibrations.

Click and drag the four dots at the left of the menu bar to tear it from the main window as a floating palette. Click and drag the floating palette back into the main window to re-dock it.

Toolbar

Toolbar buttons provide one-click shortcuts to common tasks. Buttons are gray when those functions are not currently available.

Figure 29. uProcess Toolbar Buttons.

New (ctrl-n) Create a new automated process

Open (ctrl-o) Open an existing automated process

Save (ctrl-s) Save the current automated process

Cut (ctrl-x) While editing an automated process, delete selected text and place it on the clipboard

Copy (ctrl-c) While editing an automated process, copy selected text to the clipboard

Paste (ctrl-v) While editing an automated process, paste text from the clipboard

Print (ctrl-p) Print the text of an automated process

About uProcess Find the uProcess version number. This window also contains links to the electronic version of this User Manual, the LabSmith web site, and LabSmith tech support.

Run/Pause/Quit Control the progress of an automated process.

Stop all devices Stop all activity of all uProcess devices.

Re-scan devices Check for all connected uProcess devices.

Online Establish communication between the EIB and computer.

Tip: You can also use keyboard shortcuts, shown in parentheses in the list above, to perform the commands.

To toggle toolbar buttons on or off, click the down arrow on the right side of the toolbar, then choose **Add/Remove Buttons>Standard**. Choose **Reset Toolbar** to show all buttons.

Click and drag the four dots at the left side of the toolbar to tear it from the main window as a floating palette. Click and drag the floating palette back into the main window to re-dock it.

Choose View>Toolbars and Docking Windows>Standard to show or hide the toolbar.

Tooltips

Each toolbar button (and many other buttons in uProcess) includes tooltip text that appears when you position your cursor over the button. Tooltips explain a tool's function. Choose **View>Verbose Tooltips** to view more extensive descriptions (Figure 30).

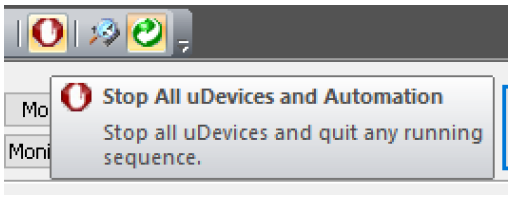


Figure 30. Tooltips Provide Helpful Tool Descriptions.

Customizing the Dashboard

Show/Hide Dashboard Elements

From the View menu choose Toolbars and Docking Windows. From this menu option you can choose to show or hide the toolbar, uDevices Window/Sequencer Window, and Output Window.

Auto-Minimize Windows

Click the "push-pin" button at the upper right of the uDevices Window/Sequencer Window or the Output Window to minimize the window to the edge of the screen. Roll over a minimized window to make it reappear; it will automatically minimize when you roll the cursor off of the window. The push-pin button will be horizontal in this "auto-minimize" mode. Click the button again to turn it vertically and disable auto-minimizing.

Detach, Dock, and Re-Position Windows

Click and drag the top of the uDevices Window/Sequencer Window or the Output Window to detach it from the Dashboard and create a floating window. Drag the floating window over the Dashboard to dock it. Positioning guides will appear as you do so to help you attach the window at the top, bottom, left, or right of the uProcess Dashboard (Figure 31).

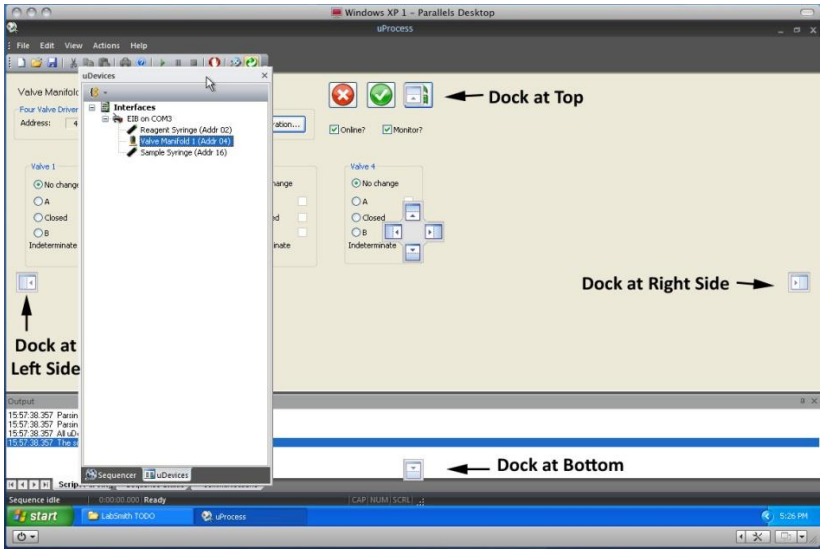


Figure 31. Positioning Guides for Re-docking a Window.

APPENDIX B. MAINTENANCE

Updating Firmware

The firmware for all uProcess devices can be field updated as follows:

Download the latest version of uProcess software from ***www.labsmith.com***.

Choose **Actions>Firmware>Update All Firmware**. uProcess will upload the latest firmware associated with the installed software to all connected syringe pumps, 4VM01 valve manifolds, and 4AM01 analog manifolds. Devices that are not attached will not be updated.

Note: AV200 series automated valves and uPS01 series pressure sensors do not have firmware and therefore do not need to be connected for firmware updates.

Note: You can update the firmware for a particular device by clicking the Configuration button in the device control panel.

uProcess devices are factory configured for optimal performance. From time to time it may be necessary to access and adjust these settings.

Valve Actuation Profile Settings

Click the **Configuration** button to open the **Configure Device** dialog box shown in Figure 32. The ***Actuation profile settings*** control the motor actuation duration and current. These settings may need to be adjusted as your valves and motors age. However, the user should **c**ontact LabSmith before changing any of the Actuation Profile Settings, as incorrect settings can cause excessive wear of the motor or prevent the motor from reaching its set possition.

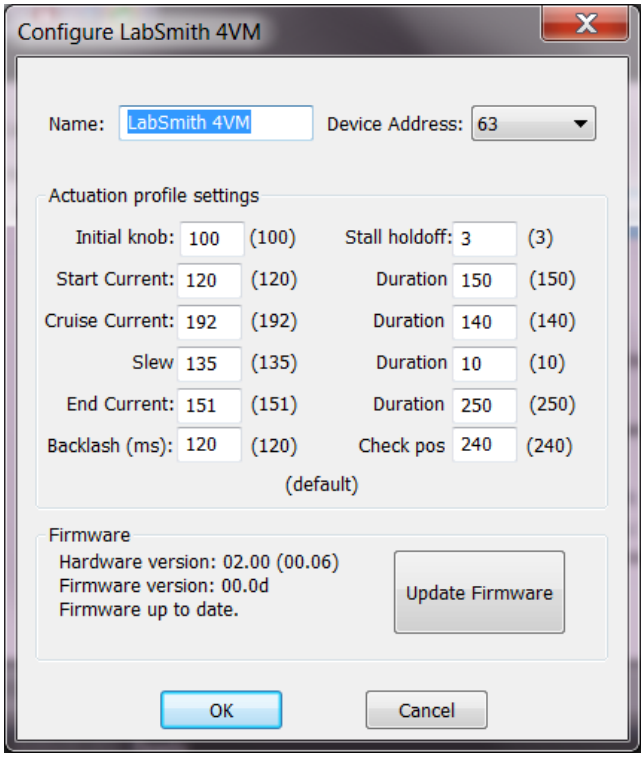
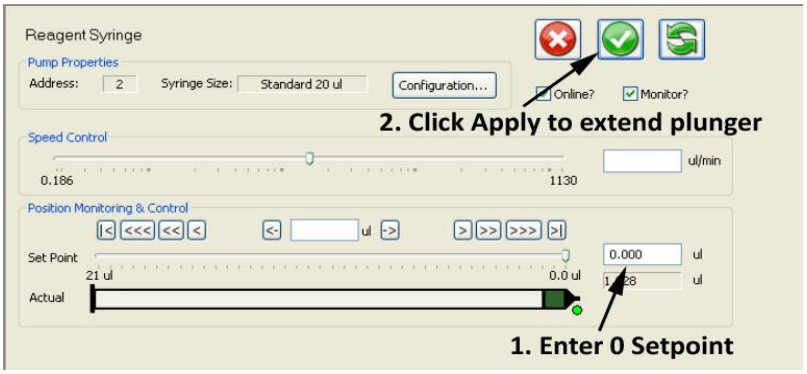


Figure 32. Configure Device Dialog Box for a 4VM01 Valve Manifold.

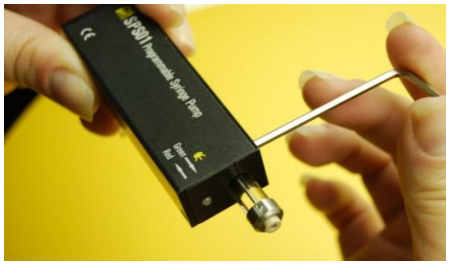
Changing the Syringe Pump Glass Assembly

The SPS01 syringe glass and plunger can be removed from the pump body to be cleaned or exchanged with a different syringe size. To remove the syringe glass and plunger:

1. Supply power to the syringe pump.
2. In the uProcess **uDevices Window**, select the syringe.
3. Enter a **Set Point** of 0 to extend the syringe plunger to the 0 volume position.



4. Click **Offline** to take the syringe offline.
5. Unplug the syringe from the breadboard.
6. Using the supplied hex wrench, loosen the screw located on the side of the syringe.



7. Gently pull the syringe glass straight out of the syringe body, being careful not to bend the syringe plunger.
8. While holding the wide end of the plunger installation wrench, insert the wrench around the plunger tip and into the syringe until it engages with the hex head at the base of the syringe plunger.

CAUTION: Be very careful not to damage Teflon end of the plunger. Damage to the plunger can cause the syringe to leak.

9. Turn the wrench counterclockwise approximately two revolutions to unscrew the plunger.
10. Gently pull the wrench and plunger out of the syringe.

    CAUTION: Do not operate the syringe without a plunger installed. Doing so can damage the syringe.

To install the syringe glass and plunger:

1. While holding the wide end of the plunger installation wrench, insert the plunger assembly into the wrench such that the hexagonal feature on the plunger is engaged with the wrench.
2. Gently insert the plunger assembly into the syringe and screw onto the shaft until hand tight (approximately two revolutions).
3. Carefully press the tapered glass syringe over the plunger assembly and into the syringe until it is seated in the syringe body.
4. Use the hex wrench to tighten the screw located on the top side of the syringe.
5. Gently pull on the syringe glass to ensure it is secure in the syringe body.
6. The glass size value must be input in the software each time you change the glass size to allow uProcess to display and compute correct volumes and volume flow rates. To select the correct glass size:
7. Click on the device in the uDevices Window.
8. Select Configuration.
9. Choose the syringe pump glass size from the **Size** drop-down list.
10. Follow instructions for calibrating, below.

    CAUTION: Ensure you are using the correct size plunger for your syringe glass.

Calibrating the Syringe Pump

Each SPS01 syringe pump is calibrated before shipment. However, each glass/plunger assembly may have slight length variations (up to 1%), and there can also be variations in the placement of the glass in the syringe body.

Each time the syringe glass is removed or moved the syringe pump should be recalibrated. Calibrating the syringe pump allows uProcess to find the plunger stopping point, thereby minimizing the dead volume within the syringe and allowing correct calculation of the syringe volume.

Note: Calibrating the syringe will run the syringe to its maximum extents and will update its calibration values. The procedure may take up to 1 minute.

To calibrate the syringe:

Select the device in the uDevices Window.

Select Configuration to open the Syringe Pump Configuration dialog box (Figure 33).

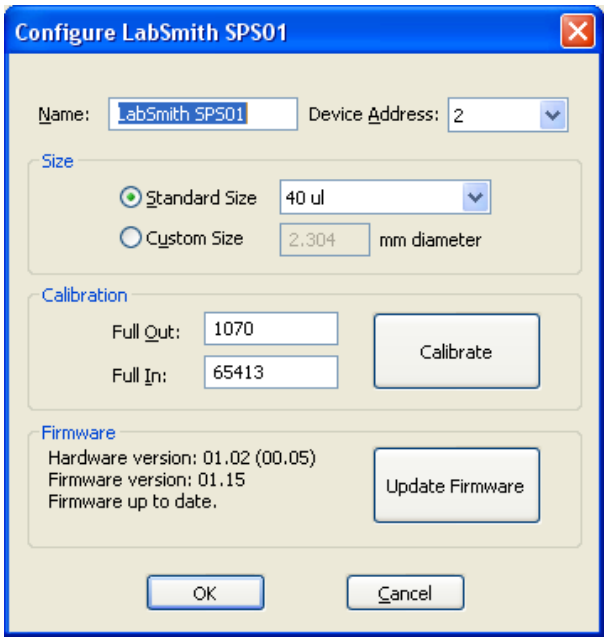


Figure 33. Syringe Pump Configuration Dialog Box.

Click the **Calibrate** button.

Select **OK** to store the values.

Tip: To obtain the maximum volume of the syringe pump. Remove the syringe glass from the SPS01 (but leave in the plunger) and run the calibration. Then replace the syringe glass, carefully pushing the syringe glass back until it hits the end of the plunger. Rerun the calibration after tightening the syringe glass in place.

Care of the Syringe Glass and Plunger

Plunger Care

Do not apply excessive force to move a plunger. Too much pressure can irretrievably bend the plunger or easily crack the syringe's glass barrel.

Avoid touching the plunger with your fingers. Abrasions, scratches, or oil due to handling the plunger with your fingers may interfere with proper plunger operation.

To minimize the chance of damage or contamination, do not store the plunger outside the glass. Use caution when reinserting the plunger into the barrel using the plunger installation tool.

For best results, dip the PTFE plunger tip into your solvent to re-wet it prior to reinserting the plunger into the barrel.

Cleaning, Sterilizing, Autoclaving, and Disinfecting

Cleaning

To clean syringes, it is best to use solvents known to be effective in solvating the sample. Preferred cleaning agents are non-alkaline, non-phosphate and non-detergent based. A biodegradable, non-phosphate, organic Cleaning Concentrate is available from Hamilton Company (Part No. 18311, www.hamiltoncompany.com).

Rinse the syringe after use with an appropriate solvent or cleaning agent. Following the use of a cleaning agent, rinse the syringe with deionized water and finally acetone. Wipe the exterior surfaces of the syringe barrel and needle dry with a lint-free tissue. Make sure that there is no residual cleaning agent in the syringe before using or storing the syringe. Do not soak or submerse the entire syringe in any cleaning agent. Prolonged contact with cleaning agents may damage bonded parts.

Sterilizing

Syringes may be sterilized with appropriate sterilizing agents such as ethylene oxide.

Autoclaving

Syringes can be autoclaved up to a temperature of 115 ºC. However, please be aware that over time autoclaving will shorten the syringes' life expectancy. Avoid rapid temperature changes, which can lead to splitting of the syringe's glass barrel. Stainless steel expands faster than glass upon heating, so for most terminations, autoclaving strains glued surfaces and may eventually lead to adhesive deterioration and leakage. The plunger should be removed from the syringe before autoclaving.

APPENDIX C. uDEVICE CURRENT CONSUMPTION

The table below shows the approximate current consumption of each uProcess device. These values can be used to calculate the total current consumption for a process and determine whether an application can be run with a single EIB200. The power supply for the EIB200 supplies 1000 mA of current, however we recommend limiting the total current to 800 mA to account for device and power supply variations.

Device	Idle Current Consumption (mA)	Active Current Consumption (mA)
EIB200 uProcess Interface	20	20
4VM01 Valve Manifold	35	35
AV201, AV202, or AV303 Automated Valve	0	50
4AM01 Analog Manifold	45	45
uPS Pressure Sensor	1	1
SPS01 Syringe Pump	10	90

Notes:
1. Current values are for 12 V power supply
2. Uncertainty +/- 3 mA
3. Valid for use with EIB200 only (current consumption will be higher with EIB100)
4. EIB200 supplied with 12 V / 1000 mA power supply.

APPENDIX D. CAPILLARY AND TUBING INSTALLATION GUIDE

LabSmith's CapTite™ one-piece fittings enable connection of components (valves, syringe pumps, interconnects, bonded-port connectors, Luer Locks, and cartridges) to 360um capillary, 1/32" tubing, or 1/16" tubing. To install the capillary or tubing:

Cut the capillary or tubing to length. Please refer to the capillary or tubing packaging for cutting instructions.

Insert the capillary into the nut end of the one-piece fitting until the tubing extends through the tapered ferrule end approximately 2 mm.

Insert the one-piece fitting into the component and turn the fitting to finger-tighten.

CAUTION: Do not over-tighten the fitting. Fittings should only be "finger-tight" to avoid damage (the 1/8" hex wrench can be used to tighten or loosen difficult-to-reach fittings).

Gently pull on the capillary tubing to ensure that it is held securely by the fitting. If the capillary comes free, loosen the fitting and repeat steps 2 and 3.

If the fitting leaks or the capillary cannot be secured, the following troubleshooting items should be checked prior to further tightening the fitting:

The capillary may be cut too short and so it is not protruding through the end of the fitting when installed

The capillary may not have a "clean cut". An angled or jagged edge will cause the fittings to leak

Broken capillary or other debris may be stuck inside the fitting or component. Try flushing the port from the opposite side to clear the debris.

If pressures greater than 10,000 psi are required, first finger-tighten all fittings and test for leaks at low pressure. Then, use the 1/8" hex wrench to tighten the fittings approximately ¼ extra revolution.

Figure 34 shows capillaries that are properly and improperly installed:

- **Red** capillary does not protrude through the one-piece fitting. The fitting will leak, and also there is a chance of compressing and damaging the tip of the one-piece fitting during installation.
- **Yellow** capillary may properly seal, but the capillary is not fully seated in the interconnect, creating dead volume.
- **Green** capillary is properly installed.

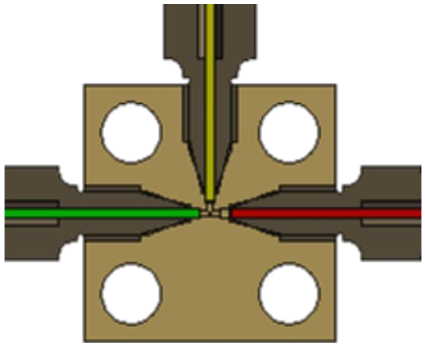


Figure 34 T-Interconnect Showing Properly Installed Capillary (Green), Improperly Seated Capillary (Yellow), and Capillary Not Protruding Through Fitting (Red).